

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 4-7: Data-link layer protocol specification – Type 7 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-7: Spécification de protocole de la couche de liaison de données –
Éléments de Type 7**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2007 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications

The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigenda or an amendment might have been published.

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee,...). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 14 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 55 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: csc@iec.ch.

A propos de l'IEC

La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC

Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études,...). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 14 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 55 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: csc@iec.ch.

INTERNATIONAL STANDARD

NORME INTERNATIONALE

**Industrial communication networks – Fieldbus specifications –
Part 4-7: Data-link layer protocol specification – Type 7 elements**

**Réseaux de communication industriels – Spécifications des bus de terrain –
Partie 4-7: Spécification de protocole de la couche de liaison de données –
Éléments de Type 7**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

PRICE CODE
CODE PRIX

XE

ICS 25.040.40; 35.100.20

ISBN 978-2-8322-1020-8

**Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.**

CONTENTS

FOREWORD.....	6
INTRODUCTION.....	7
1 Scope.....	9
1.1 General.....	9
1.2 Specifications.....	9
1.3 Procedures.....	9
1.4 Applicability.....	9
1.5 Conformance.....	9
2 Normative references	10
3 Terms, definitions, symbols and abbreviations.....	10
3.1 Reference model terms and definitions.....	10
3.2 Service convention terms and definitions.....	11
3.3 Other terms and definitions	12
3.4 Symbols and abbreviations.....	16
4 Overview of the DL-protocol	18
4.1 Overall description of medium allocation	18
4.2 Types of entities	20
4.3 Addressing	23
4.4 Flow control.....	29
4.5 Graphical representation	31
5 General structure and encoding of PhIDUs and DLPDUs and related elements of procedure.....	32
5.1 DLPDU formats and components.....	32
5.2 Description of each DLPDU component.....	32
5.3 PhIDU structure and encoding.....	36
5.4 Common DLPDU structure, encoding and elements of procedure	37
5.5 Valid DLPDU types.....	37
5.6 DLL timers.....	39
6 DLPDU-specific structure, encoding and element of procedure.....	43
6.1 General.....	43
6.2 Buffer read	43
6.3 Buffer write.....	44
6.4 Buffer transfer	44
6.5 Specified explicit request.....	45
6.6 Free explicit request.....	50
6.7 Messaging.....	53
6.8 Acknowledged messaging	58
6.9 Numbering of acknowledged messages	62
6.10 Behavior with mismatched parameters	64
7 DL-service elements of procedure, interfaces and conformance	66
7.1 General.....	66
7.2 Producer/consumer entity.....	67
7.3 Protocol elements by service.....	70
7.4 Bus arbitrator operation.....	77
7.5 Bridges.....	85
7.6 Interfaces	92

7.7 Conformance.....	94
Annex A (informative) Exemplary FCS implementation.....	97
Annex B (informative) Object modeling	99
B.1 Modeling of the IDENTIFIER object	99
B.2 Description of the IDENTIFIER object attributes	99
B.3 Modeling of the QUEUE object	103
B.4 Description of the QUEUE object attributes	103
B.5 Modeling of the BUFFER object.....	104
B.6 Description of the BUFFER object attributes.....	104
Annex C (informative) Topology of multi-segment DL-subnetwork.....	106
C.1 Introduction	106
C.2 Global specification	106
C.3 Local specification.....	107
C.4 Properties	107
C.5 Methods	107
Annex D (informative) Management of transmission errors	111
D.1 Transmission of RP_DAT_XX.....	111
D.2 Transmission of a free RP_RQ(1/2).....	111
D.3 Transmission of the specified RP_RQ1	112
D.4 Transmission of RP_MSG_NOACK.....	113
D.5 Transmission of RP_MSG_ACK.....	115
Bibliography.....	118
Figure 1 – Relationships of DLSAPs, DLSAP-addresses and group DL-addresses	13
Figure 2 – General description of medium allocation	19
Figure 3 – Internal structure of a producer/consumer entity.....	20
Figure 4 – Associated buffers and queues	22
Figure 5 – Internal structure of a bus arbitrator	23
Figure 6 – Polling BA Table	23
Figure 7 – Addressing scheme.....	24
Figure 8 – Address partitioning	26
Figure 9 – Structure of an individual physical address	27
Figure 10 – Structure of an individual logical address	27
Figure 11 – Structure of restricted physical group address.....	27
Figure 12 – Structure of a restricted logical group address	28
Figure 13 – Structure of a generalized group address	28
Figure 14 – Summary of address structure.....	29
Figure 15 – Example of an evaluation net	31
Figure 16 – Basic DLPDU structure.....	32
Figure 17 – DLPDU transmission / reception order.....	32
Figure 18 – Identifier DLPDU	38
Figure 19 – Variable response DLPDU.....	38
Figure 20 – Request response DLPDU.....	38
Figure 21 – Message response DLPDU.....	39
Figure 22 – Acknowledgement response DLPDU	39

Figure 23 – End of message transaction response DLPDU	39
Figure 24 – Buffer reading service interactions	44
Figure 25 – Buffer writing service interactions.....	44
Figure 26 – Buffer transfer service interactions.....	44
Figure 27 – Buffer transfer DLPDU sequence	45
Figure 28 – Interactions within the specified explicit request for buffer transfer service in the aperiodic window.....	46
Figure 29 – Interactions within the specified explicit request for buffer transfer service in the periodic window	47
Figure 30 – DLPDU sequence for an explicit request for a transfer	48
Figure 31 – Evaluation network for a buffer transfer specified explicit request with (RQ_INHIBITED = False).....	49
Figure 32 – Evaluation network for a buffer transfer specified explicit request with (RQ_INHIBITED = True)	49
Figure 33 – Diagram of interactions within the free explicit request for buffer transfer service.....	51
Figure 34 – Evaluation network for a free explicit request	52
Figure 35 – Diagram of interactions within the unacknowledged message transfer request service for an aperiodic transfer	55
Figure 36 – Diagram of interactions within the unacknowledged message transfer request service for a cyclical transfer.....	56
Figure 37 – DLPDU sequence for an aperiodic message transfer.....	57
Figure 38 – DLPDU sequence for a cyclical message transfer	58
Figure 39 – Diagram of interactions within the acknowledged message transfer request service for an aperiodic transfer.....	59
Figure 40 – Diagram of interactions within the acknowledged message transfer request service for a cyclical transfer	60
Figure 41 – DLPDU sequence for an aperiodic message transfer.....	61
Figure 42 – DLPDU sequence for a cyclical message transfer	62
Figure 43 – Evaluation network for message aperiodic transfer.....	65
Figure 44 – Evaluation network for message cyclic transfer	66
Figure 45 – Simplified states machine for a producer/consumer entity	67
Figure 46 – Active bus arbitrator's simplified state machine	83
Figure 47 – Typical bridge usage	85
Figure 48 – Architectural placement of bridges in OSI Basic Reference Model (ISO/IEC 7498).....	85
Figure 49 – Representation of an extended link communication	86
Figure 50 – Evaluation network for reception of an RP_MSG_ACK DLPDU.....	91
Figure 51 – Evaluation network for reception of an RP_MSG_NOACK DLPDU.....	92
Figure A.1 – Example of FCS generation	97
Figure A.2 – Example of FCS syndrome checking on reception.....	97
Figure D.1 – Evaluation DL-subnetwork for transmission of RP_DAT_XX.....	111
Figure D.2 – Evaluation DL-subnetwork for transmission of a free RP_RQ(1/2).....	112
Figure D.3 – Evaluation DL-subnetwork for transmission of the specified RP_RQ1	113
Figure D.4 – Evaluation DL-subnetwork for transmission of RP_MSG_NOACK, first behavior	114

Figure D.5 – Evaluation DL-subnetwork for transmission of RP_MSG_NOACK, second behavior	115
Figure D.6 – Evaluation DL-subnetwork for transmission of RP_MSG_ACK, first behavior	116
Figure D.7 – Evaluation DL-subnetwork for transmission of RP_MSG_ACK, second behavior	117
Table 1 – Individual and group address encoding	26
Table 2 – DLPDU control-field coding	33
Table 3 – Correspondence between name and coding of 8 bits in the control field	34
Table 4 – FCS length, polynomial and expected residual	35
Table 5 – DL-Timers	41
Table 6 – Bus arbitrator state transition table	84
Table 7 – Bridge object description	87
Table 8 – Channel object description	88
Table 9 – Segment directory object description	89
Table 10 – Network directory object description	89
Table 11 – Service primitives by type	93
Table 12 – Conformance classes	96

INTERNATIONAL ELECTROTECHNICAL COMMISSION

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-7: Data-link layer protocol specification – Type 7 elements

FOREWORD

- 1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
- 2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
- 3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
- 4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
- 5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
- 6) All users should ensure that they have the latest edition of this publication.
- 7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
- 8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
- 9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

NOTE Use of some of the associated protocol types is restricted by their intellectual-property-right holders. In all cases, the commitment to limited release of intellectual-property-rights made by the holders of those rights permits a particular data-link layer protocol type to be used with physical layer and application layer protocols in Type combinations as specified explicitly in the IEC 61784 series. Use of the various protocol types in other combinations may require permission from their respective intellectual-property-right holders.

International Standard IEC 61158-4-7 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation.

This first edition and its companion parts of the IEC 61158-4 subseries cancel and replace IEC 61158-4:2003. This edition of this part constitutes an editorial revision.

This edition of IEC 61158-4 includes the following significant changes from the previous edition:

- a) deletion of the former Type 6 fieldbus, and the placeholder for a Type 5 fieldbus data link layer, for lack of market relevance;
- b) addition of new types of fieldbuses;
- c) division of this part into multiple parts numbered -4-1, -4-2, ..., -4-19.

This bilingual version (2013-09) corresponds to the monolingual English version, published in 2007-12.

The text of this standard is based on the following documents:

FDIS	Report on voting
65C/474/FDIS	65C/485/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

The French version of this standard has not been voted upon.

This publication has been drafted in accordance with ISO/IEC Directives, Part 2.

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date indicated on the IEC web site under <http://webstore.iec.ch> in the data related to the specific publication. At this date, the publication will be:

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

NOTE The revision of this standard will be synchronized with the other parts of the IEC 61158 series.

The list of all the parts of the IEC 61158 series, under the general title *Industrial communication networks – Fieldbus specifications*, can be found on the IEC web site.

The contents of the corrigendum of January 2014 have been included in this copy.

INTRODUCTION

This part of IEC 61158 is one of a series produced to facilitate the interconnection of automation system components. It is related to other standards in the set as defined by the “three-layer” fieldbus reference model described in IEC/TR 61158-1.

The data-link protocol provides the data-link service by making use of the services available from the physical layer. The primary aim of this standard is to provide a set of rules for communication expressed in terms of the procedures to be carried out by peer data-link entities (DLEs) at the time of communication. These rules for communication are intended to provide a sound basis for development in order to serve a variety of purposes:

- a) as a guide for implementors and designers;
- b) for use in the testing and procurement of equipment;
- c) as part of an agreement for the admittance of systems into the open systems environment;
- d) as a refinement to the understanding of time-critical communications within OSI.

This standard is concerned, in particular, with the communication and interworking of sensors, effectors and other automation devices. By using this standard together with other standards positioned within the OSI or fieldbus reference models, otherwise incompatible systems may work together in any combination.

INDUSTRIAL COMMUNICATION NETWORKS – FIELDBUS SPECIFICATIONS –

Part 4-7: Data-link layer protocol specification – Type 7 elements

1 Scope

1.1 General

The data-link layer provides basic time-critical messaging communications between devices in an automation environment.

This protocol provides communication opportunities to all participating data-link entities

- a) in a synchronously-starting cyclic manner, according to a pre-established schedule, and
- b) in a cyclic or acyclic asynchronous manner, as requested each cycle by each of those data-link entities.

Thus this protocol can be characterized as one which provides cyclic and acyclic access asynchronously but with a synchronous restart of each cycle.

1.2 Specifications

This standard specifies

- a) procedures for the timely transfer of data and control information from one data-link user entity to a peer user entity, and among the data-link entities forming the distributed data-link service provider;
- b) the structure of the fieldbus DLPDUs used for the transfer of data and control information by the protocol of this standard, and their representation as physical interface data units.

1.3 Procedures

The procedures are defined in terms of

- a) the interactions between peer DL-entities (DLEs) through the exchange of fieldbus DLPDUs;
- b) the interactions between a DL-service (DLS) provider and a DLS-user in the same system through the exchange of DLS primitives;
- c) the interactions between a DLS-provider and a Ph-service provider in the same system through the exchange of Ph-service primitives.

1.4 Applicability

These procedures are applicable to instances of communication between systems which support time-critical communications services within the data-link layer of the OSI or fieldbus reference models, and which require the ability to interconnect in an open systems interconnection environment.

Profiles provide a simple multi-attribute means of summarizing an implementation's capabilities, and thus its applicability to various time-critical communications needs.

1.5 Conformance

This standard also specifies conformance requirements for systems implementing these procedures. This part of this standard does not contain tests to demonstrate compliance with such requirements.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61158-2 (Ed.4.0), *Industrial communication networks – Fieldbus specifications – Part 2: Physical layer specification and service definition*

IEC 61158-3-7, *Industrial communication networks – Fieldbus specifications – Part 3-7: Data link service definition – Type 7 elements*

ISO/IEC 7498-1, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*

ISO/IEC 7498-3, *Information technology – Open Systems Interconnection – Basic Reference Model: Naming and addressing*

ISO/IEC 10731, *Information technology – Open Systems Interconnection – Basic Reference Model – Conventions for the definition of OSI services*

3 Terms, definitions, symbols and abbreviations

For the purposes of this document, the following terms, definitions, symbols and abbreviations apply.

3.1 Reference model terms and definitions

This standard is based in part on the concepts developed in ISO/IEC 7498-1 and ISO/IEC 7498-3, and makes use of the following terms defined therein.

3.1.1 correspondent (N)-entities	[7498-1]
correspondent DL-entities (N=2)	
correspondent Ph-entities (N=1)	
3.1.2 DL-address	[7498-3]
3.1.3 DL-connection	[7498-1]
3.1.4 DL-connection-end-point	[7498-1]
3.1.5 DL-connection-end-point-identifier	[7498-1]
3.1.6 DL-name	[7498-3]
3.1.7 DL-protocol	[7498-1]
3.1.8 DL-protocol-connection-identifier	[7498-1]
3.1.9 DL-protocol-control-information	[7498-1]
3.1.10 DL-protocol-data-unit	[7498-1]
3.1.11 DL-relay	[7498-1]
3.1.12 DL-service-connection-identifier	[7498-1]
3.1.13 DL-service-data-unit	[7498-1]

3.1.14 DL-user-data	[7498-1]
3.1.15 flow control	[7498-1]
3.1.16 (N)-entity	[7498-1]
DL-entity	
Ph-entity	
3.1.17 (N)-interface-data-unit	[7498-1]
DL-service-data-unit (N=2)	
Ph-interface-data-unit (N=1)	
3.1.18 (N)-layer	[7498-1]
DL-layer (N=2)	
Ph-layer (N=1)	
3.1.19 (N)-service	[7498-1]
DL-service (N=2)	
Ph-service (N=1)	
3.1.20 (N)-service-access-point	[7498-1]
DL-service-access-point (N=2)	
Ph-service-access-point (N=1)	
3.1.21 (N)-service-access-point-address	[7498-1]
DL-service-access-point-address (N=2)	
Ph-service-access-point-address (N=1)	
3.1.22 peer-entities	[7498-1]
3.1.23 Ph-interface-control-information	[7498-1]
3.1.24 Ph-interface-data	[7498-1]
3.1.25 primitive name	[7498-3]
3.1.26 reset	[7498-1]
3.1.27 responding-DL-address	[7498-3]
3.1.28 routing	[7498-1]
3.1.29 segmenting	[7498-1]
3.1.30 sequencing	[7498-1]
3.1.31 system management	[7498-1]
systems-management	

3.2 Service convention terms and definitions

This standard also makes use of the following terms defined in ISO/IEC 10731 as they apply to the data-link layer:

- 3.2.1 confirm (primitive);**
 requestor.deliver (primitive)
- 3.2.2 deliver (primitive)**
- 3.2.3 DL-service-primitive;**
 primitive
- 3.2.4 DL-service-provider**

3.2.5 DL-service-user

3.2.6 DL-user-optional-facility

3.2.7 indication (primitive)
acceptor.deliver (primitive)

3.2.8 multi-peer

3.2.9 request (primitive);
requestor.submit (primitive)

3.2.10 requestor

3.2.11 response (primitive);
acceptor.submit (primitive)

3.2.12 submit (primitive)

3.3 Other terms and definitions

NOTE Many definitions are common to more than one protocol Type; they are not necessarily used by all protocol Types.

For the purpose of this part of IEC 61158, the following definitions also apply:

3.3.1
acknowledgement response DLPDU

information that the recipient of an acknowledged message emits in order to signal either the proper reception of the message or the lack of available resources to store the message, received by the DLE on the local link that emitted the message which requested the acknowledgement

3.3.2
basic cycle

sequence of scanning by the bus-arbitrator of:

- a) a set of DLCEP-identifiers for variables, requests, and cyclical application messages,
- b) plus the window provided for aperiodic exchanges,
- c) plus the window provided for message services,
- d) plus the window provided for synchronization

3.3.3
basic transaction

succession of DLPDUs related to a single DL-service instance

3.3.4
bus-arbitrator (BA)

DLE that controls each data producer's right to access the medium

NOTE At any given instant one and only one bus-arbitrator is active in each DL-segment of a DL-subnetwork.

3.3.5
consumed identified variable

identified variable that corresponds to a DLCEP-identifier for which the entity in question receives data

3.3.6
control field

portion of an emitted or received DLPDU that gives the nature of the data exchanged and the type of exchange

3.3.7**destination address**

three octets specifying the DL-segment of the DLE to whom the message is sent, and the destination DLSAP's sub-address within the local link DL-segment

3.3.8**DLCEP-address**

information that the bus-arbitrator emits to allocate the medium to a data producer for the purpose of exchanging a variable

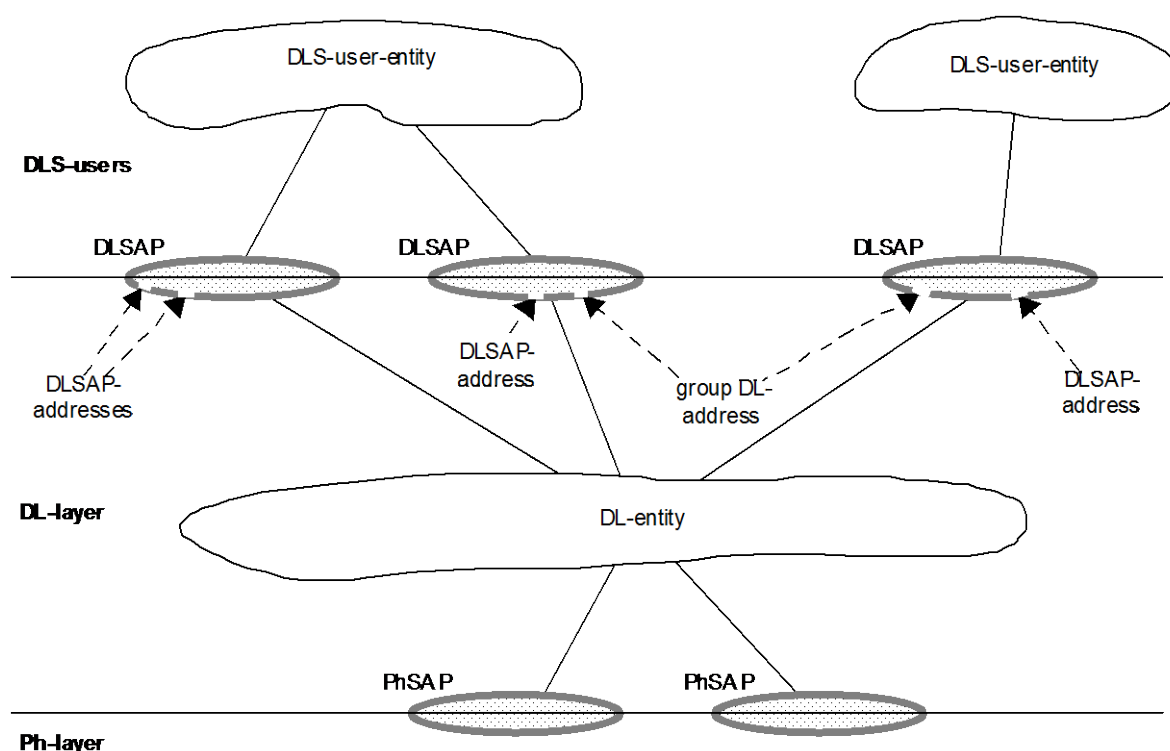
3.3.9**DL-segment, link, local link**

single DL-subnetwork in which any of the connected DLEs may communicate directly, without any intervening DL-relaying, whenever all of those DLEs that are participating in an instance of communication are simultaneously attentive to the DL-subnetwork during the period(s) of attempted communication

3.3.10**DLSAP**

distinctive point at which DL-services are provided by a single DL-entity to a single higher-layer entity.

NOTE This definition, derived from ISO/IEC 7498-1, is repeated here to facilitate understanding of the critical distinction between DLSAPs and their DL-addresses. (See Figure 1.)



NOTE 1 DLSAPs and PhSAPs are depicted as ovals spanning the boundary between two adjacent layers.

NOTE 2 DL-addresses are depicted as designating small gaps (points of access) in the DLL portion of a DLSAP.

NOTE 3 A single DL-entity may have multiple DLSAP-addresses and group DL-addresses associated with a single DLSAP.

Figure 1 – Relationships of DLSAPs, DLSAP-addresses and group DL-addresses

3.3.11**DL(SAP)-address**

either an individual DLSAP-address, designating a single DLSAP of a single DLS-user, or a group DL-address potentially designating multiple DLSAPs, each of a single DLS-user.

NOTE This terminology is chosen because ISO/IEC 7498-3 does not permit the use of the term DLSAP-address to designate more than a single DLSAP at a single DLS-user

3.3.12

(individual) DLSAP-address

DL-address that designates only one DLSAP within the extended link

NOTE A single DL-entity may have multiple DLSAP-addresses associated with a single DLSAP.

3.3.13

end of message transaction indication DLPDU

information that the source entity of a message emits in order to return link access control to the bus-arbitrator at the end of a message transaction

3.3.14

extended link

DL-subnetwork, consisting of the maximal set of links interconnected by DL-relays, sharing a single DL-name (DL-address) space, in which any of the connected DL-entities may communicate, one with another, either directly or with the assistance of one or more of those intervening DL-relay entities

NOTE An extended link may be composed of just a single link.

3.3.15

frame

denigrated synonym for DLPDU

3.3.16

group DL-address

DL-address that potentially designates more than one DLSAP within the extended link. A single DL-entity may have multiple group DL-addresses associated with a single DLSAP. A single DL-entity also may have a single group DL-address associated with more than one DLSAP

3.3.17

identified variable (or simply "variable")

DLL variable (buffer) for which an associated DLCEP-identifier has been defined

3.3.18

identifier

16-bit word associated with a system variable. A DLCEP-identifier uniquely designates a single variable within the DL-subnetwork

3.3.19

invalid DLCEP-identifier

identifier not recognized locally

3.3.20

local link

set of devices that respect the DL-protocol and that are interconnected through a medium . Only one bus-arbitrator is active on a single local link

3.3.21

macrocycle

set of basic cycles needed for all cyclical DLCEP-identifiers to be scanned

3.3.22

message DL-address

information that the bus-arbitrator emits to allocate the medium to a source entity for a message transfer

3.3.23**message response DLPDU**

information that a data producer emits in response to a message DLCEP-identifier DLPDU

NOTE The desired destination entity or entities pick up this information.

3.3.24**node**

single DL-entity as it appears on one local link

3.3.25**periodic scanning of variables**

action by the bus-arbitrator that guarantees the cyclical exchange of variables

NOTE This is the basic principle of the Type 7 DL-service and protocol.

3.3.26**produced identified variable**

identified variable that corresponds to a DLCEP-identifier for which the DLE emits data

3.3.27**receiving DLS-user**

DL-service user that acts as a recipient of DL-user-data

NOTE A DL-service user can be concurrently both a sending and receiving DLS-user.

3.3.28**request DLCEP-address**

information that the bus-arbitrator emits to allocate the medium to the initiator of an explicit request for a variable exchange

3.3.29**request response DLPDU**

information that the initiator of an explicit request for a variable exchange emits in response to a request DLCEP-identifier DLPDU, and to whose transmittal the bus-arbitrator also responds

3.3.30**sending DLS-user**

DL-service user that acts as a source of DL-user-data

3.3.31**source address**

24-bit word including the DL-segment number of the entity sending the message, and the entity's sub-address within the DL-segment

3.3.32**timers**

see 5.6

3.3.33**triggered message scanning**

function of a bus-arbitrator that makes it possible to transfer messages

3.3.34**triggered scanning of variables**

function of a bus-arbitrator that makes possible the non-cyclical exchange of variables

3.3.35

triggered periodic scanning of messages

function of a bus-arbitrator that makes it possible to request triggered message exchanges cyclically

3.3.36

triggered periodic scanning of variables

function of a bus-arbitrator that makes it possible to request triggered variable transfers cyclically

3.3.37

turnaround time

time interval between reception or emission of the last MAC symbol of a DLPDU, signaled by a SILENCE indication from the PhL, and the reception or emission of the first MAC symbol of the subsequent DLPDU, signaled by an ACTIVITY indication from the PhL, both as measured in a given station

3.3.38

variable response DLPDU

information that a data producer emits in response to a DLCEP-identifier DLPDU, which also alerts data consumers to the relevance of the immediately time-proximate DLPDU

3.4 Symbols and abbreviations

3.4.1 BA	bus-arbitrator
3.4.2 B_Dat_Cons	buffer which contains the value of the data consumed
3.4.3 B_Dat_Prod	buffer which contains the value of the data produced
3.4.4 B_Req1/2	buffer containing the list of DLCEP-identifiers that are the objet of a specified explicit request for a transfer at the priority 1 (urgent) or 2 (normal)
3.4.5 ID_DAT	DLPDU used to allocate the medium to a buffer transfer
3.4.6 ID_MSG	DLPDU used to allocate the medium to a message exchange
3.4.7 ID_RQ1/2	DLPDU used to allocate the medium to a request for a buffer transfer
3.4.8 PRT	physical reaction time
3.4.9 Q_IDMSG	queue of requested DLCEP-identifiers received by the BA for message transfer
3.4.10 Q_IDRQ1/2	queue of requested DLCEP-identifiers received by the BA at the priority 1 (urgent) or 2 (normal)
3.4.11 Q_Msg_Cyc	queue which contains messages to be emitted that are associated with cyclical exchanges
3.4.12 Q_Msg_Aper	queue which contains messages to be emitted that are associated with aperiodic exchanges
3.4.13 Q_Req1/2	queue containing the list of DLCEP-identifiers that are the objet of a free explicit request for a transfer at the priority 1 (urgent) or 2 (normal)
3.4.14 Q_RPRQ	queue for aperiodic transfers in progress
3.4.15 RQ_Inhibit	Indicator used to manage explicit request for variable

exchanges

- | | |
|--------------------------------|--|
| 3.4.16 RP_ACK | DLPDU used to transfer an acknowledgement of message exchange, Transfer OK |
| 3.4.17 RP_DAT | DLPDU used to carry the value of the identified variable previously requested. |
| 3.4.18 RP_DAT_MSG | DLPDU used to carry the value of the identified variable previously requested with a request for a message exchange. |
| 3.4.19 RP_DAT_RQ1/2 | DLPDU used to carry the value of the identified variable previously requested with a request for an explicit transfer of variables. |
| 3.4.20 RP_DAT_RQ1/2_MSG | DLPDU used to carry the value of the identified variable previously requested with a request for an explicit transfer of variables and a request for a message transfer. |
| 3.4.21 RP_MSG_ACK | DLPDU used to carry the message to be exchanged with a request of acknowledgement of this exchange. |
| 3.4.22 RP_MSG_NOACK | DLPDU used to carry the message to be exchanged without a request of acknowledgement of this exchange. |
| 3.4.23 RP_NAK | DLPDU used to transfer an acknowledgement of message exchange, that the message was received correctly, but was not stored by the DLE. |
| 3.4.24 RP_END | DLPDU used to terminate a message exchange. |
| 3.4.25 STT | station turnaround time |
| 3.4.26 TI | latency time |

4 Overview of the DL-protocol

4.1 Overall description of medium allocation

An element known as the **bus-arbitrator (BA)** controls the right of each data producer to access the medium by emitting a DLPDU containing a DLCEP-identifier. At any given instant there should be only one active bus-arbitrator in each local link.

NOTE The term "data producer" designates the sole station connected to the local link that is recognized as having the responsibility of emitting the data associated with the identifier DLPDU circulating on the medium.

Each transaction belongs to one of the three medium allocation classes defined below:

- cyclical exchange of variables, requests, or messages,
- explicit request for variable exchange,
- explicit request for message transfer.

For cyclical variable exchanges a basic transaction is made up of the following phases. The bus-arbitrator broadcasts a variable identifier DLPDU. The sole producer of the information required then broadcasts a variable response DLPDU. During this phase consumers take the information from the local link. Figure 2 shows the various phases of a variable exchange transaction. When one transaction has been completed the bus-arbitrator begins the following transaction according to guidelines defined when the system is configured.

During a cyclical variable exchange the information producer can, using the response DLPDU, transmit to the BA an explicit request for the exchange of variables or messages.

For an explicit request for a variable exchange, a basic transaction is made up of the following phases: The bus-arbitrator broadcasts a request identifier DLPDU. Then the initiator of the request broadcasts a request response DLPDU. One or more transactions identical to the cyclical variables exchange transaction then follow.

For message transfers a basic transaction consists of the following phases: The bus-arbitrator broadcasts a message identifier DLPDU. Then the message response DLPDU is exchanged between the communicating entities. This DLPDU may or may not be followed by an acknowledgement DLPDU. The source entity then transmits an end of transaction indication DLPDU to the bus-arbitrator.

The time interval separating the reception or emission of the end of one DLPDU and the emission or reception of the following DLPDU is known as the station's turnaround time, whether the station's function be that of a producer/consumer or bus-arbitrator.

A more detailed definition of turnaround time is given in 3.3.37. The impact of turnaround time on data-link timers is described in 5.6.2.

The role of the bus-arbitrator is to "give the floor" to each data producer, taking into account the services required for the type of data (according to the three medium allocation classes defined above).

The bus-arbitrator thus has three types of functions:

- periodic scanning of variables or periodic triggering of variable and message exchanges,
- triggered scanning of variables,
- triggered scanning of messages.

In addition, the bus-arbitrator can provide a synchronization function in order to guarantee the constant length of scanning cycles.

Each type of scanning takes place in a specific window, that is, respectively in a periodic window, an aperiodic variables window, an aperiodic messages window, or a synchronization window, as shown in Figure 2. These four windows constitute a basic scanning cycle.

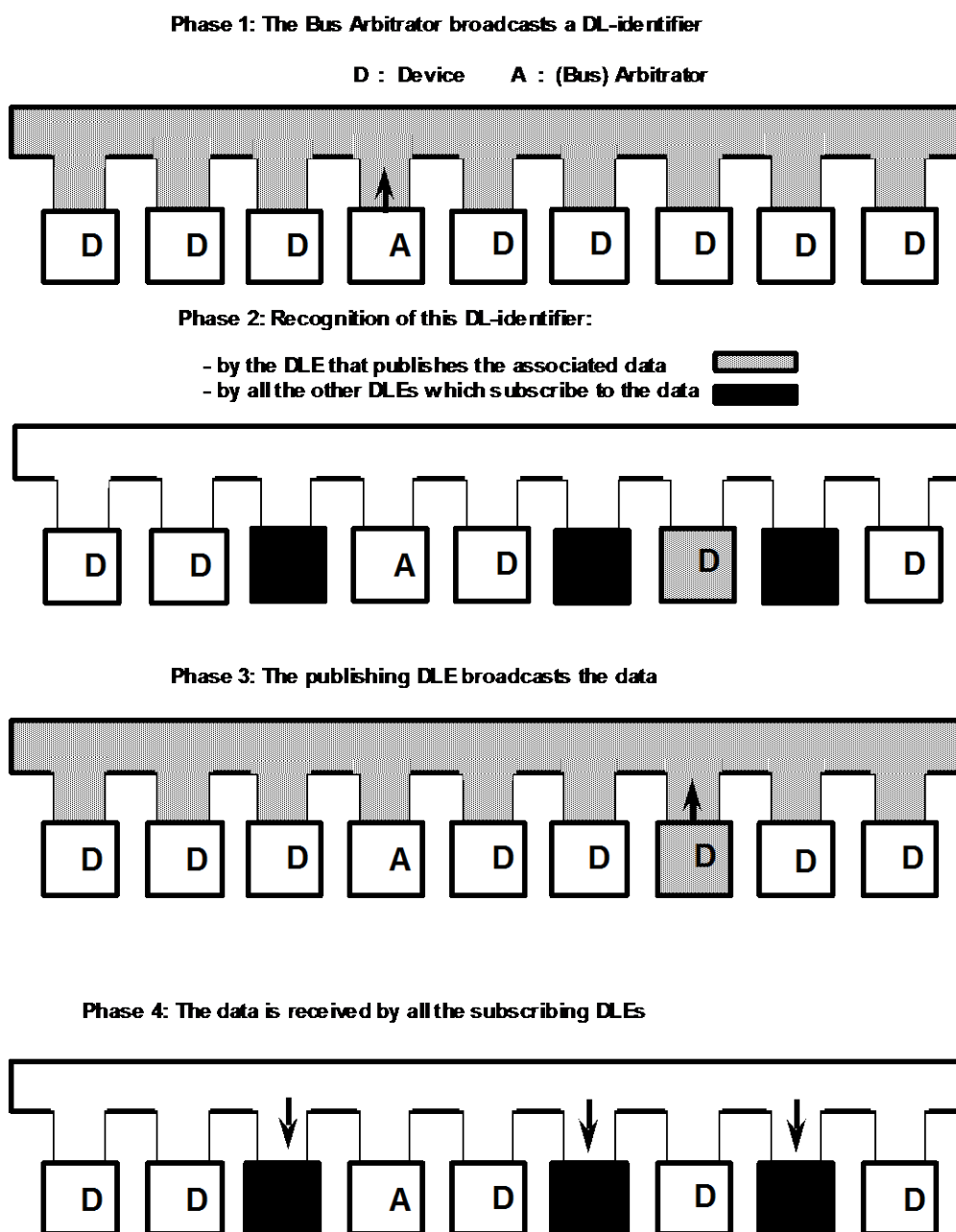


Figure 2 – General description of medium allocation

The medium access technique has the following characteristics:

- broadcasting of identified variables,
- increased efficiency in cyclical variable exchanges,
- parameters for medium sharing can be set by the user when the system is configured,
- guaranteed access time for cyclical variable exchanges, under all circumstances and regardless of the number of requests for triggered variable and/or message exchanges.
- possibility of triggering a transaction in accordance with a global clock, that is a clock that indicates the same time for all stations.

- give cyclical exchanges highest priority,
- respect the scanning period associated with each variable,
- give different priorities to triggered messages transfers and variable exchanges. These transactions are triggered in adjustable windows: the lengths of the "aperiodic variable" and "aperiodic message" windows are defined in terms of maximum limits set by the user when the system is configured.
- change the priority of aperiodic transactions by inserting them in the periodic window.

4.2.1 Producer/consumer entity

NOTE 5.5 describes all types of DLPDUs and also explains the use of various fields.

- a mechanism for analyzing DLPDU type (use of the control field),
- a table of identifiers recognized upon emission,
- a table of identifiers recognized upon reception (both tables are defined at the interface between the DLL and system management),

a mechanism that provides read/write access to variables and messages.

[illegible]

The following are associated with each identifier produced:

- for cyclical message
- Documents**
- Document Sharing Hub

- an indicator of the validity of the identifier produced (system management),
- for each of the B_DATprod and B_REQ buffers, an availability indicator,
- an indicator associated with Q_MSGcyc: queue filled or not,
- indicators used to manage explicit requests for variable exchange and message transfer requests:
 - explicit request for buffer transfer in progress (RQ),
 - priority associated with an explicit request (PR),
 - scope of the explicit request (RQ_INHIBIT)
- message transfer request in progress (MSG) if the identifier has been reserved for aperiodic message transfer,
- reference to the queue of messages to be emitted associated with aperiodic message transfer.

The following are associated with each identifier consumed:

- a buffer called B_DATcons, which contains the value of the variable consumed,
- an indicator of the validity of the identifier consumed (system management),
- an indicator linked to the management of B_DATcons: availability of the buffer (access conflict),

NOTE The buffer availability indicator provides information concerning the integrity of data manipulated by service primitives.

Each entity that supports a message transfer request service also uses:

- a queue called Q_MSGaper, which is associated with aperiodic message transfer and contains messages to be emitted,
- a queue called Q_MSGrec that contains messages received.
- an indicator of queue status (full or not) is associated with the queue reserved for aperiodic message transfer.

In addition, each entity that supports acknowledged message service has the following characteristics:

- value of the source entity's even/odd bit,
- maximum value of the restart counter,
- current value of the restart counter.

The following are associated with the queue Q_MSGrec:

- an indicator of queue status (full or not),
- value of the destination entity's even/odd bit,
- value of the stored source address.

If the entity also supports the free explicit request for buffer transfer service, there are two global queues for holding free explicit requests for buffer transfer:

- Q_REQ1 is associated with urgent requests,
- Q_REQ2 is associated with normal requests.

A status indicator (full or not) is associated with each queue.

Figure 4 shows many of these associated buffers and queues.

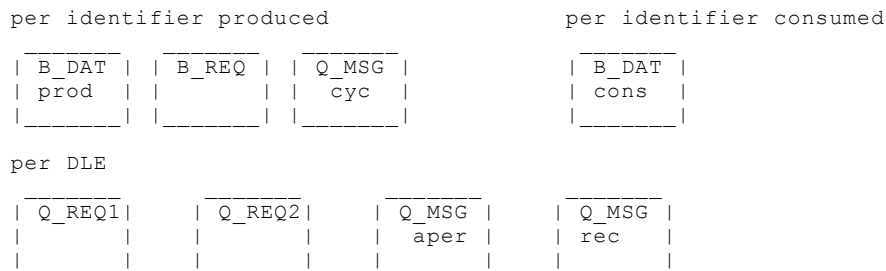


Figure 4 – Associated buffers and queues

Timers associated with the data-link protocol are also managed by the producer/consumer entity (see 5.6).

Annex B presents an object model that defines the attributes of a DLCEP-identifier.

4.2.2 Bus arbitrator entity

The function provided by the bus-arbitrator consists of "giving permission to speak" to each data producer. This permission is granted for three types of scanning:

- periodic or triggered periodic scanning of variables and messages,
- triggered scanning of variables,
- triggered scanning of messages.

The bus-arbitrator also provides the following functions:

- chaining of basic transactions,
- chaining of the various types of scanning,
- analysis of DLPDU control fields (the control field carries aperiodic variable and message requests),
- filling of aperiodic windows in accordance with these requests.

NOTE A basic transaction is made up of the succession of DLPDUs related to a single service.

In addition, the bus-arbitrator can provide a synchronization function to guarantee the constant length of a basic scanning cycle.

Each type of scanning takes place in a special window: respectively in a periodic window, an aperiodic variables window, an aperiodic messages window, or a synchronization window.

The four windows constitute a basic scanning cycle.

Figure 5 provides an overview of the bus-arbitrator structure. A more complete description of the bus-arbitrator is given in 7.4.

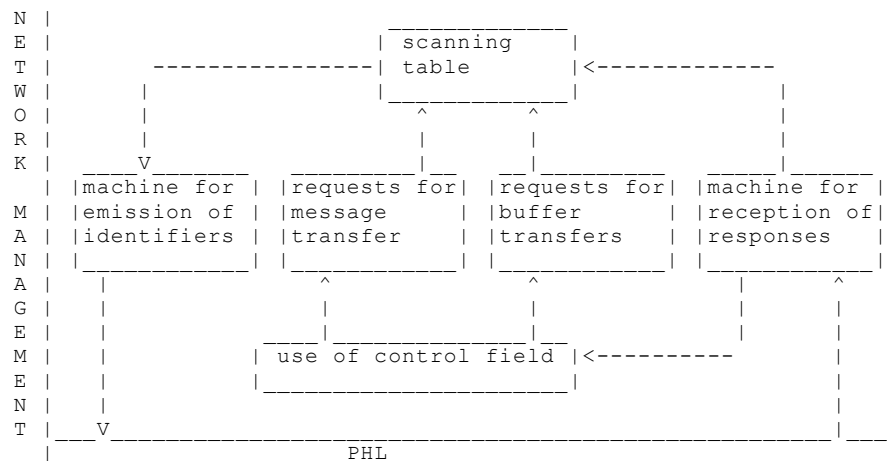


Figure 5 – Internal structure of a bus arbitrator

The conceptual model of the bus-arbitrator details the various tables and queues needed to provide the services offered by the DLL.

The bus-arbitrator manages according to system configuration, as shown in Figure 6:

- a scanning table with static portions and portions that are modified dynamically during scanning,
- a recovery buffer for the triggered periodic scanning of variables,
- a queue called Q_IDRQ1 for urgent explicit requests for buffer transfer,
- a queue called Q_IDRQ2 for normal explicit requests for buffer transfer,
- a queue for aperiodic transfers in progress (Q_RPRQ),
- a queue called Q_IDMSG for requests for message transfer,
- a basic padding sequence used to fill the synchronization window.

NOTE The management algorithm and/or calculation of these tables should provide equal access rights for the various entities, that is, starving certain entities should be avoided.

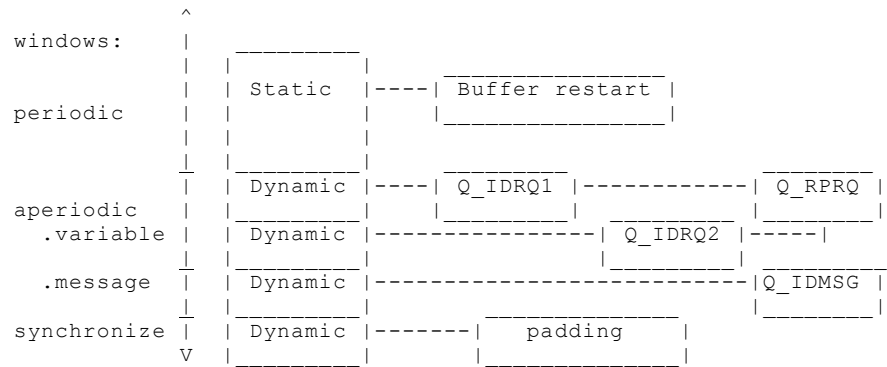


Figure 6 – Polling BA Table

4.3 Addressing

4.3.1 DLSAP-user relationship

The role of the DLL is to transmit information reliably and in accordance with a transmission protocol.

Within a single physical station several user entities have access to DLL services. Each of these entities uses a service access point (SAP) belonging to the DLL, in conformance with the ISO's static model in which each entity (N+1) is recognized by the address of the service access point (N) to which the entity is statically attached.

Several associations between user entities are possible on the level of access points to DLL services.

Thus, in accordance with the ISO model, a user entity (N+1) requests the establishment of a connection (N) in order to communicate with another user entity (N+1). The (N) entities, each of which is linked to one of the two (N+1) entities, create and manage this connection after negotiating it.

All exchanges between the entities (N+1) thus associated take place using this connection. The connection is identified locally at each SAP by a specific connection end point identifier (CEPi).

The number of DLSAPs corresponds to the number of entities in a station that are potential users of DLL services.

The number of CEPi per DLSAP corresponds to the number of potential communication links allocated to the (N+1) entity linked to the DLSAP.

4.3.2 OSI model relationship

This DLL follows the same rules as ISO/IEC 7498. Each potential user entity calls upon the services of the DLL through a DLSAP.

The DLL places two distinct addressing spaces as the disposal of the DLS-user:

- the addressing space concerning buffer transfer, each identifier being coded in 16 bits,
- the addressing space concerning message exchanges, which enables addressing messaging DLS-users, each address being coded in 24 bits.

The identifiers, encoded in 16 bits, allow addressing buffers within a local link, whereas the messaging addresses, encoded in 24 bits are meant to identify all the messaging DLS-users of the DL-subnetwork, that is those of all the DL-segments of the DL-subnetwork. Consequently, a communication system composed of n local links will have a single messaging address space but will have n identifier spaces to address the buffers.

Figure 7 illustrates this usage.

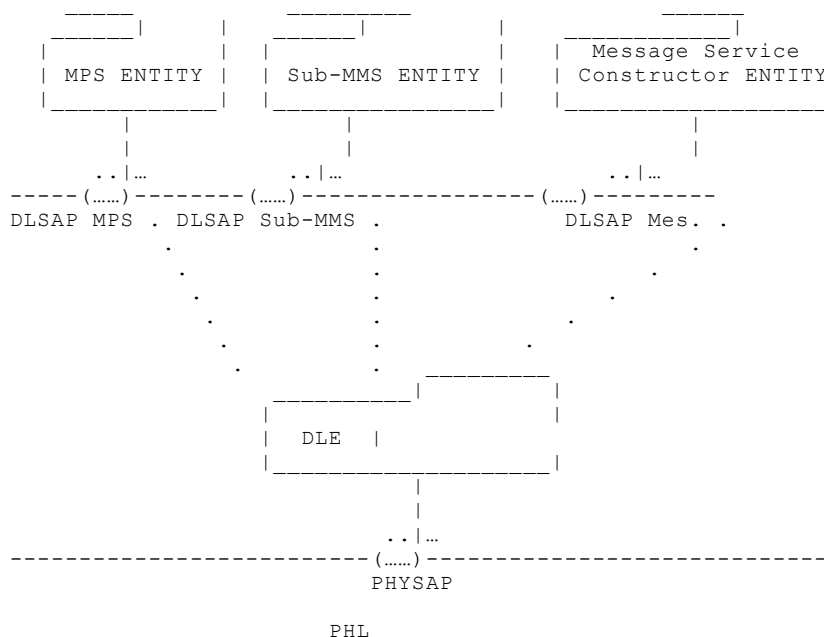


Figure 7 – Addressing scheme

4.3.2.1 Buffer transfer addressing

For reasons relating to performance and determinism, all associations between the various DLL user entities are known and defined when the system is configured.

Thus upon configuration the number of DLSAP is known, as well as the number of connections within the DLSAP.

This DLL uses broadcasting as its mode of transmission over the physical support. There is one producer and one or more consumers of each piece of information transmitted.

The connections within each DLSAP are multipoint connections providing unilateral communication (connections have more than two extremities, and over these connections data communication occurs in a single direction set in advance: from the producer to the consumers).

Connection end point identifiers (CEPi) that are known as identifiers and are encoded using 16 bits identify associations between user entities.

Thus CEPis are not recognized locally within a DLSAP, but globally throughout the DL-subnetwork, and the role of the DLSAP address is less important in the addressing model.

Thus the concept of an DLSAP is not the same as in the general ISO model. This DLL represents a group of between 1 and n identifiers employed by a user entity attached to the DLL. On the other hand, a DLSAP does not address the Application entity directly, but through the use of identifiers.

Each connection (identifier) can be used differently (according to the configuration and periodic or aperiodic services chosen) to communicate with other stations (variable transfers) or with the bus-arbitrator (requests for aperiodic transfers).

4.3.2.2 Message exchange addressing

4.3.2.2.1 Addressing requirements

The addressing of the messaging DLS-users must allow meeting the usual communication requirement such as indicated in the ISO and which are as follows:

- point-to-point communication between two DLS-users, whether or not located in the same local link,
- multipoint communication between an entity and all the entities of a group, the latter being located in the same equipment or the local link or the extended link,
- communication by distribution between a DLS-user and all the other DLS-users of a DLE, a local link or the extended link.

NOTE This corresponds to the reservation of a particular group containing all the entities of a device, a local link or the extended link. The addressing of the messaging DLS-users must distinguish between two types of addresses:

- the physical address, in order to identify an application entity with respect to its physical location, by means of the DL-segment number and the number of the station where it is located,
- the logic address, in order to identify an application entity whose physical location will not be required.

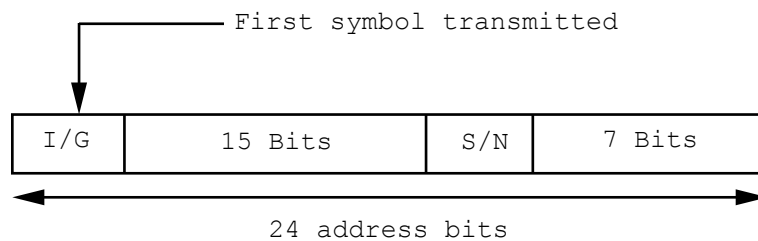
The notions of restricted groups on the DL-subnetwork with respect to groups of DLS-users in a device or in a DL-segment, have been defined with an aim to optimize the flow on the bus. Thus when addressing a group, it is thus necessary to have the possibility of a filter on the level of each bridge. The function of this filter is to prevent distribution on all the DL-segments of the extended link when this is not necessary.

The addressing of the buffers limits the number of devices on a DL-segment to 256. This must be taken into account in the messaging addressing mode.

4.3.2.2.2 Encoding of the addresses

The source and addressee DLS-users exchanging messages are identified in a DL-subnetwork with the help of addresses which are found in the link protocol data units (RP_MSG_NOACK and RP_MSG_ACK).

In order to meet the individual addressing requirements of the DLS-users in a device, in a group of devices of a local link or a group of devices of the extended link, the addressing space offered by the 24 bits is divided as shown in Figure 8.



I/G designates an individual DLS-user (I) or a group of DLS-users (G)

S/N designates a group of DLS-users on a DL-segment (S) or on the DL-subnetwork (N)

Figure 8 – Address partitioning

The encoding of the individual addresses and the groups of addresses is shown in Table 1.

Table 1 – Individual and group address encoding

I/G	S/N	Type of addressing
0	0	Individual address
1	0	Address of a group in a DL-segment
not significant	1	Address of a group in the DL-subnetwork

The link layer thus offers an addressing space which is divided into two sub-spaces:

- one containing link addresses meant to identify DLS-users individually,
- the other meant to identify DLS-user groups.

The group addresses are themselves divided into two sub-spaces:

- the sub-space defining addresses of restricted groups, thus enabling identification of the DLS-users which belong to the same local link,
- the other space defining the addresses of generalized groups, thus enabling identification of the application identities which all belong globally to the DL-subnetwork.

Each of these previously defined sub-spaces (individual addresses and addresses of restricted groups) are divided into:

- physical addresses,
- logical addresses.

4.3.2.2.2.1 Individual addressing

The individual addresses enabling a DLS-user to correspond with one and only one other DLS-user are structured as shown in Figure 9 and Figure 10:

a) Individual physical addresses:

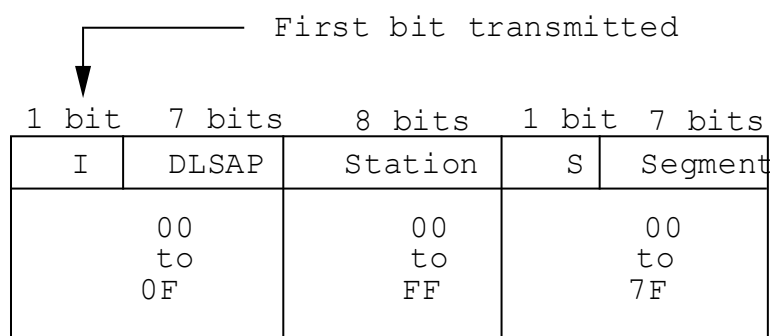


Figure 9 – Structure of an individual physical address

This structure allows identifying 16 physical DLSAPs per device.

b) Individual logical addresses:

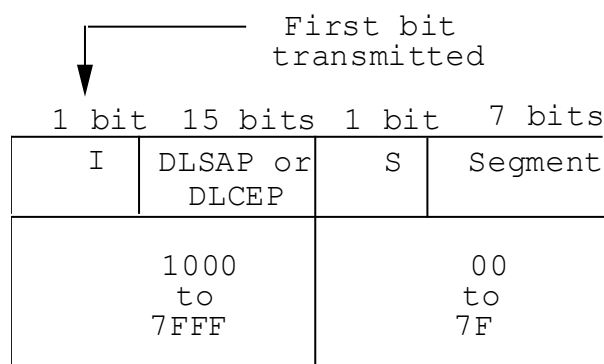


Figure 10 – Structure of an individual logical address

The addressing capacity is a total of 28K logical DLSAPs and DLCEPs per DL-segment.

4.3.2.2.2 Addressing of restricted groups

The restricted group addresses which enable a DLS-user to correspond with a group of DLS-users of the same device or of the same DL-segment are represented by their codes as shown in Figure 11 and Figure 12:

a) Addresses of restricted physical groups:

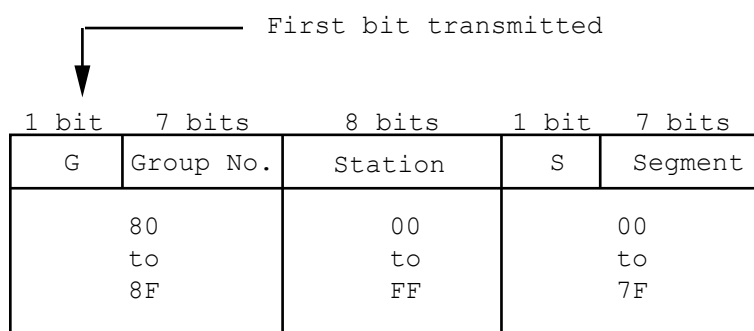


Figure 11 – Structure of restricted physical group address

The addressing capacity is then 16 numbers of physical restricted groups in a DLE. The distribution in a device is ensured by the specific group number, 8F.

b) Addresses of restricted logical groups:

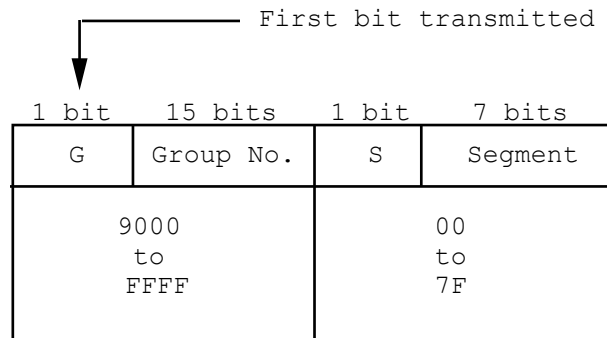


Figure 12 – Structure of a restricted logical group address

This structure allows addressing 28K numbers of restricted logical groups in a local link. FFFF corresponds to the distribution group in a DL-segment.

4.3.2.2.3 Addressing of generalized groups

The addresses of generalized groups allow a DLS-user to correspond with a group of DLS-users on the extended link. These addresses are divided into three sub-areas:

- an addressing sub-area which can be used by all the equipment of the extended link,
- a reserved sub-area,
- a vendor addressing sub-area.

The encoding structure of these addresses is as shown in Figure 13:

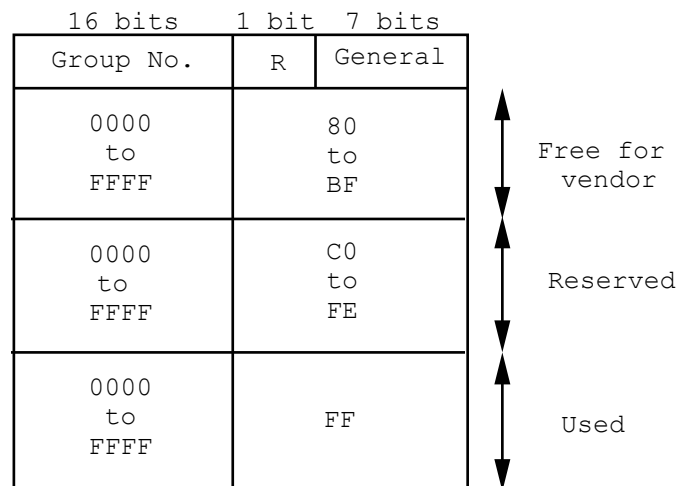


Figure 13 – Structure of a generalized group address

This addressing area allows identifying generalized 64K groups by sub-area element.

The distribution to the entire extended link is ensured by means of the particular group number FFFF.

4.3.2.3 Addressing capacity

The encoding rules offer the following possibilities for DL-subnetwork in terms of maximum capacity:

- 126 DL-segments in a DL-subnetwork;
- the "00" segment number is the DL-segment number by default;
- 256 DLEs per DL-segment in a DL-subnetwork;
- 16 physical DLSAPs in a DLE;
- 16 physical restricted group DL-addresses in a DLE;
- 28K logical DLSAPs in a local link;
- 28K logical restricted group DL-addresses in a local link;
- 64K generalized groups DL-addresses in the DL-subnetwork in the used sub-area.

The exact structure of the encoding of the beginning of an RP_MSG_XX message response DLPDU is as shown in Figure 14:

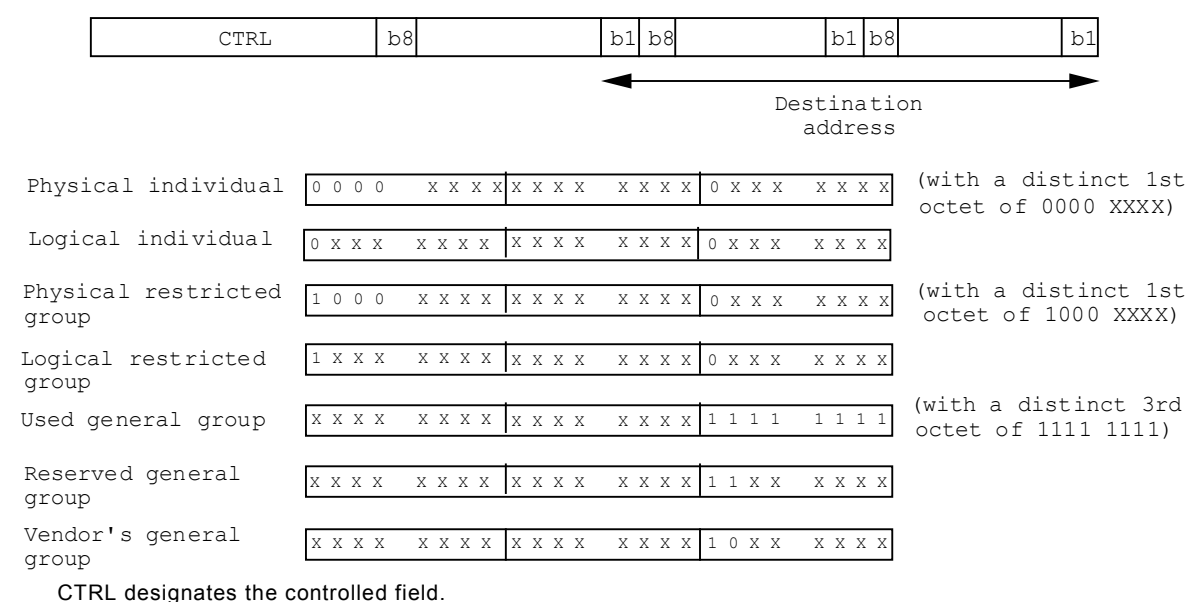


Figure 14 – Summary of address structure

The source address follows the destination address according to the same encoding.

4.4 Flow control

4.4.1 Variable exchanges

For the cyclical transfer of identified variables the principle of periodic scanning implies the replacement of an old variable value with a new variable independent of any reading access by the data consumer.

Management of flow control for these exchanges is definitively settled upon configuration:

- the bus-arbitrator manages flow control by respecting the scanning period associated with the variable,
- stations manage flow control by associating a buffer with each identified variable.

This principle also applies to triggered variable transfers since the user is only interested in the most recent validated value, and since the user associates a buffer with each identified variable. In addition, upon configuration of a DLL system a time delimiter is defined for the aperiodic window for triggered variable transfers for each of the bus-arbitrator's basic scanning cycles.

4.4.2 Message transfers

Flow control for message transfers is handled by the acknowledgement mechanism and by the queues for messages to be emitted and messages received.

When a DLL system is configured a time delimiter is defined for the aperiodic window for triggered message transfers for each of the bus-arbitrator's basic scanning cycles.

For acknowledged message transfers a portion of flow control is provided upon emission since the DLL only accepts requests for acknowledged message transfers if there is available space in the queue for messages to be emitted that is specified in the request. With this type of transfer there is also flow control upon reception by the queue of messages received and through the transmission of an acknowledgement DLPDU.

4.4.3 Detection of DLPDU duplication/loss

Detection mechanisms provided apply to errors caused by communication problems (cut-off DLPDUs, FCS error) or by out-of-service stations.

These errors include:

- loss of a DLCEP-identifier DLPDU,
- lack of a response DLPDU (DLPDU lost or station absent),
- loss of a request response DLPDU,
- loss of a message response DLPDU,
- loss of a message acknowledgement DLPDU,
- loss of a DLPDU indicating the end of a message transaction.

and can cause either the loss or the duplication of data.

DLPDU losses are taken into account in the graphs of final state machines defining data-link protocols.

Loss or duplication of DLPDUs can have the following impact on services:

- loss of a DLCEP-identifier DLPDU or a response concerning periodically exchanged variables is not usually detrimental since the value of the variable will be broadcast again during the next local link cycle;
- loss of an unacknowledged message DLPDU or of a DLPDU associated with a triggered variable exchange will only be handled by the upper layers;
- loss of an acknowledgement DLPDU results in a negative confirmation being sent to the initiating DLS-user;
- the concept of DLPDU duplication does not apply to periodically exchanged variables since the principle of cyclical variable refreshing implies that the same variable will be sent repeatedly;
- acknowledged message DLPDUs are protected from duplication by an even/odd numbering mechanism for messages;
- the duplication of a DLPDU associated with an explicit request for a buffer transfer will result in an additional overriding of the variable. This duplication is not detrimental since the concept of exchanging variables cyclically or upon explicit request is based on an asynchronous refreshing that is handled by the local link, and that the consumer cannot control. The semantics of data transmitted must be compatible with this working principle, that is, the variables exchange service is designed for the transmission of statuses, and not for the transmission of sequences of events.

4.5 Graphical representation

To be helpful for the reader some mechanisms are explained by a graphical representation. These representations are based on evaluation network built on an oriented graph with three types of nodes:

- the states, represented by a circle,
- the requests represented by a rectangle,
- the transitions, represented by a horizontal line.

The evaluation network is built according to the following principles, as illustrated in Figure 15:

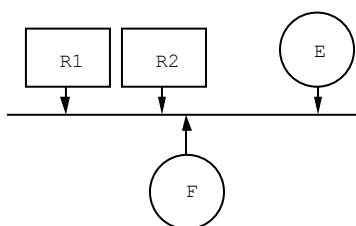


Figure 15 – Example of an evaluation net

When the described system is in state E, the arrival of request R1 or R2 results in a transition which switches it state F.

On the other hand, crossing a transition takes place, by convention, in zero time.

4.5.1 Simple actions

Simple actions can be associated with each transition, corresponding either to the transmission of requests or to the execution of local actions.

They are materially represented by a triangle, labeled with the name of the request or of the description of the executed action.

4.5.2 Conditions

The crossing of a transition can concern a condition. In this case, the oriented graph associated with the representation, shows a condition associated with the arc which precedes the transition.

A corollary of this representation allows defining choices for crossing a transition from the same initial state.

In the same way as for a simple transition, the transmission of an action can be associated with the crossing of a conditional transition.

5.2.3 Control and addressing field

The control and addressing field specifies the type of DLPDU being exchanged:

- bit 1 of the control field indicates whether the DLPDU is a DLCEP-identifier DLPDU (bit 1 = 1) or a response DLPDU (bit 1 = 0)
- bits 2 through 7 of the control field, where each bit has a specific function, as shown in Table 2:
 - if bit 2 is set at 1 the DLPDU is related to a buffer transfer
 - if bit 3 is set at 1 the DLPDU is related to a message transfer
 - if bit 4 is set at 1 the DLPDU is related to an explicit request for a buffer transfer
 - if bit 5 is set at 1 the DLPDU is related to an acknowledgement
 - bit 6 indicates priority or the result of the acknowledgement
 - bit 7 set at 1 is used only to indicate an end of message transaction DLPDU;
- bit 8 of the control field (the first bit transmitted) is used only for a message response DLPDU or an acknowledgement response DLPDU. Bit 8 provides for even/odd numbering of messages to support duplicate message detection.

For a DLCEP-identifier DLPDU these bits specify the type of exchange in progress. For a response DLPDU these bits indicate the type of exchange in progress as well as requests for message transfer and explicit requests for variable exchanges, with their priorities.

Table 2 – DLPDU control-field coding

Bits	Identifier DLPDU
2 3 4 5 6 7	
1 0 0 0 0 x	allocation for identified variable
0 1 0 0 0 x	allocation for message
0 0 1 0 1 x	allocation for urgent explicit request
0 0 1 0 0 x	allocation for normal explicit request
	Response DLPDU
1 0 0 0 0 x	identified variable response
1 1 0 0 0 x	identified variable response + message request
1 0 1 0 1 x	identified variable response + explicit urgent request
1 0 1 0 0 x	identified variable response + explicit normal request
1 1 1 0 1 x	identified variable response + explicit urgent request + message request
1 1 1 0 0 x	identified variable response + explicit normal request + message request
0 1 0 1 0 x	acknowledged message
0 1 0 0 0 x	unacknowledged message
0 0 0 1 1 x	positive acknowledgement
0 0 0 1 0 x	negative acknowledgement
0 0 1 0 1 x	list of identifiers for an urgent request
0 0 1 0 0 x	list of identifiers for a normal request
0 0 0 0 1 x	end of message transaction

For identifier DLPDUs the control and addressing field is extended to 8 + 16 bits. In this case it includes a single DL-identifier that represents a local-link DL-address encoded in 16 bits.

For request response DLPDUs the control and addressing field is extended to 8 + 16 times 16 bits. In this case it includes the list of DL-identifiers associated with the variables requested.

For message response DLPDUs the control and addressing field is extended to 8 + 24 + 24 bits. In this case it includes two extended-link DL-addresses — the destination and source, respectively, of the message.

Table 3 gives the correspondence between the name and coding of the control field's bits.

Table 3 – Correspondence between name and coding of 8 bits in the control field

Bits	Associated name
1 2 3 4 5 6 7 8	
1 1 0 0 0 0 x x	ID_DAT
1 0 1 0 0 0 x x	ID_MSG
1 0 0 1 0 1 x x	ID_RQ1
1 0 0 1 0 0 x x	ID_RQ2
0 1 0 0 0 0 x x	RP_DAT
0 1 1 0 0 0 x x	RP_DAT_MSG
0 1 0 1 0 1 x x	RP_DAT_RQ1
0 1 0 1 0 0 x x	RP_DAT_RQ2
0 1 1 1 0 1 x x	RP_DAT_RQ1_MSG
0 1 1 1 0 0 x x	RP_DAT_RQ2_MSG
0 0 1 0 1 0 x N	RP_MSG_ACK
0 0 1 0 0 0 x x	RP_MSG_NOACK
0 0 0 0 1 1 x N	RP_ACK+
0 0 0 0 1 0 x N	RP_ACK-
0 0 0 1 0 1 x x	RP_RQ1
0 0 0 1 0 0 x x	RP_RQ2
0 0 0 0 0 0 1 x	RP_END
NOTE N takes the value 0 or 1; x values are ignored but should be 0	

5.2.4 DLS-user-data field

Only response DLPDUs include a DLS-user-data field. The DLS-user-data field contains:

- either the value of a variable,
- or a message.

5.2.5 DLPDU frame check sequence (FCS) field

Within this subclause, any reference to bit K of an octet is a reference to the bit whose weight in a one-octet unsigned integer is 2^K .

NOTE This is sometimes referred to as "little endian" bit numbering.

For the protocol Type of this standard, as in other International Standards (for example, ISO/IEC 3309, ISO/IEC 8802 and ISO/IEC 9314-2), DLPDU-level error detection is provided by calculating and appending a multi-bit frame check sequence (FCS) to the other DLPDU fields during transmission to form a "systematic code word"¹⁾ of length n consisting of k DLPDU message bits followed by $n - k$ (equal to 16) redundant bits, and by calculating during

¹⁾ W. W. Peterson and E. J. Weldon, Jr., *Error Correcting Codes* (2nd edition), MIT Press, Cambridge, 1972.

reception that the message and concatenated FCS form a legal (n,k) code word. The mechanism for this checking is as follows:

The generic form of the generator polynomial for this FCS construction is specified in equation (6) and the polynomial for the receiver's expected residue is specified in equation (11). The specific polynomials for this standard are specified in Table 4. An exemplary implementation is discussed in Annex A.

Table 4 – FCS length, polynomial and expected residual

Item	Value
$n-k$	16
$G(x)$	$X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^6 + X^3 + X^2 + X + 1$ (notes 1, 2, 3)
$R(x)$	$X^{15} + X^{14} + X^{13} + X^9 + X^8 + X^7 + X^4 + X^2$ (note 4)
NOTE 1 Code words $D(X)$ constructed from this $G(X)$ polynomial have Hamming distance 4 for lengths ≤ 344 octets and Hamming distance 5 for lengths ≤ 15 octets.	
NOTE 2 This $G(X)$ polynomial is relatively prime to all, and is thus not compromised by any, of the polynomials commonly used in DCEs (modems): the differential encoding polynomial $1 + X^{-1}$ and all primitive scrambling polynomials of the form $1 + X^{-j} + X^{-k}$.	
NOTE 3 This $G(X)$ polynomial is the optimal 16-bit polynomial for burst error detection over DLPDUs of 300 octets or less when the statistics of the error burst have a Poisson distribution (as is the usual case).	
NOTE 4 The remainder $R(x)$ should be 1110 0011 1001 0100 (X^{15} to X^0 , respectively) in the absence of errors.	

5.2.5.1 At the sending DLE

The original message (that is, the DLPDU without an FCS), the FCS, and the composite message code word (the concatenated DLPDU and FCS) shall be regarded as vectors $M(X)$, $F(X)$, and $D(X)$, of dimension k , $n - k$, and n , respectively, in an extension field over $GF(2)$. If the message bits are $m_1 \dots m_k$ and the FCS bits are $f_{n-k-1} \dots f_0$, where

$m_1 \dots m_8$ form the first octet sent,
 $m_{8N-7} \dots m_{8N}$ form the N th octet sent,
 $f_7 \dots f_0$ form the last octet sent, and
 m_1 is sent by the first PhL symbol(s) of the message and f_0 is sent by the last PhL symbol(s) of the message (not counting PhL framing information),

NOTE This "as transmitted" ordering is critical to the error detection properties of the FCS.

then the message vector $M(X)$ shall be regarded to be

$$M(X) = m_1X^{k-1} + m_2X^{k-2} + \dots + m_{k-1}X^1 + m_k \quad (1)$$

and the FCS vector $F(X)$ shall be regarded to be

$$\begin{aligned} F(X) &= f_{n-k-1}X^{n-k-1} + \dots + f_0 \\ &= f_{15}X^{15} + \dots + f_0 \end{aligned} \quad (2)$$

The composite vector $D(X)$, for the complete DLPDU, shall be constructed as the concatenation of the message and FCS vectors

$$\begin{aligned} D(X) &= M(X)X^{n-k} + F(X) \\ &= m_1X^{n-1} + m_2X^{n-2} + \dots + m_kX^{n-k} + f_{n-k-1}X^{n-k-1} + \dots + f_0 \\ &= m_1X^{n-1} + m_2X^{n-2} + \dots + m_kX^{16} + f_{15}X^{15} + \dots + f_0 \quad (\text{for the case of } k = 15) \end{aligned} \quad (3)$$

The DLPDU presented to the PhL shall consist of an octet sequence in the specified order.

The redundant check bits $f_{n-k-1} \dots f_0$ of the FCS shall be the coefficients of the remainder $F(X)$, after division by $G(X)$, of $L(X) (X^k + 1) + M(X) X^{n-k}$

where $G(X)$ is the degree $n-k$ generator polynomial for the code words

$$G(X) = X^{n-k} + g_{n-k-1}X^{n-k-1} + \dots + 1 \quad (4)$$

and $L(X)$ is the maximal weight (all ones) polynomial of degree $n-k-1$

$$\begin{aligned} L(X) &= \frac{X^{n-k} + 1}{X + 1} = X^{n-k-1} + X^{n-k-2} + \dots + X + 1 \\ &= X^{15} + X^{14} + X^{13} + X^{12} + \dots + X^2 + X + 1 \quad (\text{for the case of } k = 15) \end{aligned} \quad (5)$$

That is,

$$F(X) = L(X) (X^k + 1) + M(X) X^{n-k} \pmod{G(X)} \quad (6)$$

NOTE 1 The $L(X)$ terms are included in the computation to detect initial or terminal message truncation or extension by adding a length-dependent factor to the FCS.

NOTE 2 As a typical implementation when $n-k = 16$, the initial remainder of the division is preset to all ones. The transmitted message bit stream is multiplied by X^{n-k} and divided (modulo 2) by the generator polynomial $G(X)$, specified in equation (7). The ones complement of the resulting remainder is transmitted as the $(n-k)$ -bit FCS, with the coefficient of X^{n-k-1} transmitted first.

5.2.5.2 At the receiving DLE

The octet sequence indicated by the PhE shall be concatenated into the received DLPDU and FCS, and regarded as a vector $V(X)$ of dimension u

$$V(X) = v_1X^{u-1} + v_2X^{u-2} + \dots + v_{u-1}X + v_u \quad (7)$$

NOTE 1 Because of errors u can be different than n , the dimension of the transmitted code vector.

A remainder $R(X)$ shall be computed for $V(X)$, the received DLPDU and FCS, by a method similar to that used by the sending DLE (see 5.2.5.1) in computing $F(X)$

$$\begin{aligned} R(X) &= L(X) X^u + V(X) X^{n-k} \pmod{G(X)} \\ &= r_{n-k-1}X^{n-k-1} + \dots + r_0 \end{aligned} \quad (8)$$

Define $E(X)$ to be the error code vector of the additive (modulo-2) differences between the transmitted code vector $D(X)$ and the received vector $V(X)$ resulting from errors encountered (in the PhS provider and in bridges) between sending and receiving DLEs.

$$E(X) = D(X) + V(X) \quad (9)$$

If no error has occurred, so that $E(X) = 0$, then $R(X)$ will equal a non-zero constant remainder polynomial

$$R_{ok}(X) = L(X) X^{n-k} \pmod{G(X)} \quad (10)$$

whose value is independent of $D(X)$. Unfortunately $R(X)$ will also equal $R_{ok}(X)$ in those cases where $E(X)$ is an exact non-zero multiple of $G(X)$, in which case there are “undetectable” errors. In all other cases, $R(X)$ will not equal $R_{ok}(X)$; such DLPDUs are erroneous and shall be discarded without further analysis.

NOTE 2 As a typical implementation, the initial remainder of the division is preset to all ones. The received bit stream is multiplied by X^{n-k} and divided (modulo 2) by the generator polynomial $G(X)$, specified in equation (7).

5.3 PhIDU structure and encoding

Each PhIDU consists of Ph-interface-control-information (PhICI) and in some cases one octet of Ph-interface-data. When the DLE transmits a DLPDU, it computes a DLPDU check sequence for the DLPDU as specified in 5.2.5, concatenates the DLPDU and DLPDU check sequence, and transmits the concatenated pair as a sequence of PhIDUs as follows:

- The DLE issues a single Ph-DATA request primitive with PhICI specifying start-of-activity, and awaits the consequent Ph-DATA confirm primitive.

- b) The DLE issues a sequence of Ph-DATA request primitives with PhICI specifying DATA, each accompanied by one octet of the DLPDU as Ph-interface-data, from first to last octet of the DLPDU, and after each Ph-DATA request primitive awaits the consequent Ph-DATA confirm primitive.
- c) The DLE issues a sequence of two Ph-DATA request primitives with PhICI specifying DATA, each accompanied by one octet of the FCS as Ph-interface-data, from first to last octet of the FCS, and after each Ph-DATA request primitive awaits the consequent Ph-DATA confirm primitive.
- d) The DLE issues a single Ph-DATA request primitive with PhICI specifying END-OF-DATA-AND-ACTIVITY, and awaits the consequent Ph-DATA confirm primitive.

The DLE forms a received DLPDU by concatenating the sequence of octets received as Ph-interface-control-information of consecutive Ph-DATA indications, computing a DLPDU check sequence for those received octets as specified in 5.2.5, and checks the syndrome of the computed FCS for correctness as follows:

- e) The DLE receives a single Ph-DATA indication primitive with PhICI specifying START-OF-ACTIVITY, and initializes its computation of an FCS for the received DLPDU.
- f) The DLE receives a sequence of Ph-DATA indication primitives with PhICI specifying DATA, each accompanied by one octet of the received DLPDU as Ph-interface-data, incrementally computes an FCS on the received octet, and concatenates all but the last two of those received octets to form the received DLPDU.
- g) The DLE receives a single Ph-DATA indication primitive with PhICI specifying either END-OF-DATA, END-OF-DATA-AND-ACTIVITY or END-OF-ACTIVITY, and checks the syndrome of the computed FCS for correctness:
 - 1) If the PhICI specified END-OF-DATA or END-OF-DATA-AND-ACTIVITY, and the computed FCS syndrome was correct, then the DLE reports the reconstructed DLPDU and the two octets of received FCS as a correctly-received DLPDU suitable for further analysis.
 - 2) Otherwise the DLE increments its management statistics to reflect the erroneously received DLPDU.

5.4 Common DLPDU structure, encoding and elements of procedure

Each DLPDU consists of a DLPDU control field which specifies the type of DLPDU and conveys small size (fractional octet) parameters of the DLPDU; zero to two explicit address fields, each containing a DL-address, all of the same length; and for most DLPDUs, a user data field conveying all or part of a DLSDU. To this is appended before transmission, and removed after reception, an FCS field used to check the integrity of the received DLPDU.

5.5 Valid DLPDU types

For easier reading, DLPDUs are described without their DLPDU start and DLPDU end sequences.

For each type of DLPDU a name is given for the control field. This name is shown in parenthesis.

The correspondence between this name and the 8-bit control field code is given in the table below.

For each type of DLPDU we have indicated the separation between fields that contain data-link protocol control information (DLPCI) and fields containing data (DLSDU).

5.5.1 Identifier DLPDU

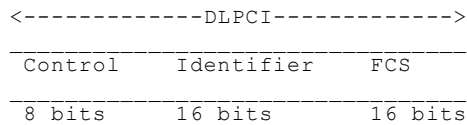


Figure 18 – Identifier DLPDU

There are four types of identifier DLPDU, whose basic structure is shown in Figure 18:

- medium allocation for identified variables (ID_DAT)
- medium allocation for message (ID_MSG)
- medium allocation for urgent explicit request (ID_RQ1)
- medium allocation for normal explicit request (ID_RQ2).

5.5.2 Variable response DLPDU

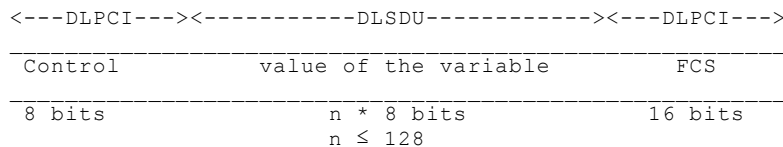


Figure 19 – Variable response DLPDU

There are six types of variable value response DLPDU, whose basic structure is shown in Figure 19:

- identified variables (RP_DAT)
- identified variables + message request (RP_DAT_MSG)
- identified variables + urgent explicit request (RP_DAT_RQ1)
- identified variables + normal explicit request (RP_DAT_RQ2)
- identified variables + urgent explicit request + message request (RP_DAT_RQ1_MSG)
- identified variables + normal explicit request + message request (RP_DAT_RQ2_MSG).

5.5.3 Request response DLPDU

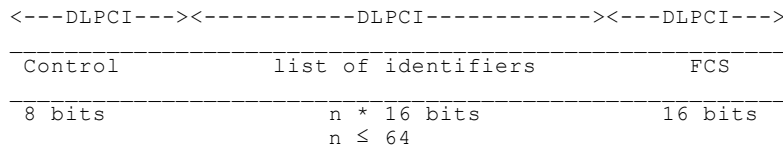


Figure 20 – Request response DLPDU

There are two types of request response DLPDU, whose basic structure is shown in Figure 20:

- list of urgent identifiers (RP_RQ1)
- list of normal identifiers (RP_RQ2).

5.5.4 Message response DLPDU

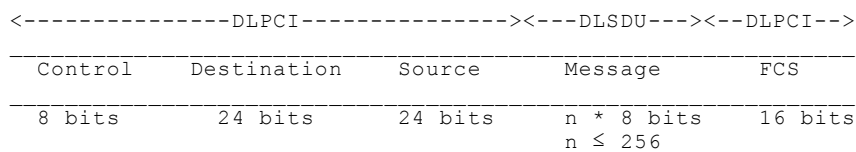


Figure 21 – Message response DLPDU

There are two types of message response DLPDU, whose basic structure is shown in Figure 21:

- acknowledged message (RP_MSG_ACK)
- unacknowledged message (RP_MSG_NOACK).

5.5.5 Acknowledgement response DLPDU

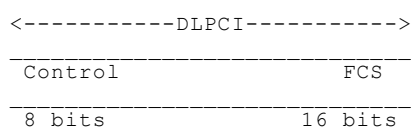


Figure 22 – Acknowledgement response DLPDU

There are two types of acknowledgement response DLPDU, whose basic structure is shown in Figure 22:

- positive acknowledgement (RP_ACK+) indicates that the message has been correctly received by the remote DLE,
- negative acknowledgement (RP_ACK-) indicates that the distant entity's queue for received message is filled.

5.5.6 End of message transaction response DLPDU

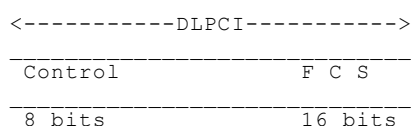


Figure 23 – End of message transaction response DLPDU

There is one type of end of message transaction response DLPDU, whose basic structure is shown in Figure 23:

NOTE The chaining of basic DLPDUs is indicated in the specifications for each service.

The RP_END DLPDU involves 2 entities: the source entity and the bus-arbitrator. The source entity returns control of the local link to the bus-arbitrator at the end of a message transaction.

5.6 DLL timers

This subclause briefly describes the timers involved in the description of data-link final state devices, proposes relationships between the various timers, and describes the importance of a station's turnaround time in the definition of its timers.

5.6.1 Common criteria

The criterion common to all stations for launching timers is the absence of activity on the local link.

Two signals are used to evaluate this criterion: "Absence of activity" (SILENCE) and "Signal present" (BUSY).

A reference timer T0 is defined as corresponding to the maximum silent time permitted on a local link.

The T0 timer is a global systems parameter.

T0 is activated upon reception of an indication of the absence of activity. This indication is emitted by the Type 2 PhL (the primitive PhL-DATA indication (SILENCE)).

T0 is deactivated by any correct functioning of the local link, upon reception of the indication of signal presence. This indication is emitted by the Type 2 PhL (the primitive PhL-DATA indication (BUSY)).

The maximum silent time permitted on the local link, that is, T0 is linked to the turnaround time of stations in the system.

5.6.2 Turnaround time

The reception or emission of the last MAC symbol is signaled by a SILENCE indication from the PhL. Likewise, the reception or emission of the first MAC symbol is signaled by an BUSY indication from the PhL.

The time lapsed between these two signals within a single station defines the station turnaround time (STT), whether the station's function be that of producer/consumer or bus-arbitrator.

Turnaround time takes into account parameters that are local to the station and parameters that are associated with the system to which the station belongs. These parameters are:

- physical reaction time (PRT): the time the PhL needs to detect the end or beginning of activity. These times depend on physical local link parameters:
 - station: time needed to detect the absence or presence of a signal
 - support: time needed to traverse the media and active copper or optical stars
- latency time (TI) resulting from processing time in the station.

TI takes on various values depending on the protocol situation in which the station finds itself:

- a) the station has emitted a DLPDU and will emit the next DLPDU,
- b) the station has received a DLPDU and will emit the next DLPDU,
- c) the station has emitted a DLPDU and will receive the next DLPDU,
- d) the station has received a DLPDU and will receive the next DLPDU.

All the parameters combine to form a minimum station turnaround time (under the most favorable conditions) and a maximum station turnaround time (under the least favorable conditions).

For the system to function correctly the following criteria *must* be respected:

- Rule 1)** a station that receives a DLPDU must always have a shorter turnaround time than the station that emitted the DLPDU, regardless of the two stations (receiving station, emitting station) involved.

This condition is expressed by:

Max(i) STT receiving station $i \leq \text{Min}(j)$ STT emitting station j

Rule 2) minimum turnaround time corresponds to the maximum value of physical reaction time PRT.

Rule 3) given the impact of a station's turnaround time on transmission yield (use of the bandwidth), it is important that maximum STT take yield criteria into account.

Rules 1 and 2 are interworking criteria. Rule 3 is a performance criterion that can be verified on the level of system management through access to the maximum turnaround time value and the resulting T0 timer.

NOTE The choice of minimum and maximum STT values can result in different interoperability and performance classes.

5.6.3 Timer functions

Five timers are defined using T0 as a basis.

The T1, T4, and T6 timers correspond to the use of T0 in the specific context of the status of a DLE (Station or bus-arbitrator). Activation and deactivation conditions are thus identical (SILENCE or BUSY), and the action after expiration of the timers differs according to the status of the DLE.

The T3 and T5 timers also refer to T0: their lengths (which are different from T0) are calculated using T0 length as a basis. In addition, they are activated and deactivated under the same conditions (SILENCE or BUSY).

An additional T2 timer is also used. T2 is not based on T0. This timer determines the length of a basic synchronized cycle.

All of these timers are listed in Table 5

Table 5 – DL-Timers

Name of timer	Location	Function
<u>T1</u>	B.A.	monitors the absence of RPxx after IDxx
T2	B.A.	monitors the filling of the synchronization window by identifiers from the basic padding sequence
<u>T3</u>	B.A.	monitors the absence of IDxx after RPxx switching of bus-arbitrators
<u>T4</u>	consuming station	monitors the absence of RP_DATxx after ID_DAT
<u>T5</u>	B.A.	monitors the lack of RP_END after ID_MSG
<u>T6</u>	message source station	monitors the absence of RP_ACK after RP_MSG_ACK

Timer definitions refer to names of statuses mentioned in the protocol descriptions presented in Clause 7.

5.6.3.1 Timer T1

- Is located in the active bus-arbitrator
- Its purpose is to monitor the emission of a variable response DLPDU or a request response DLPDU within a given time.
- Is active when the bus-arbitrator is waiting for a variable response DLPDU or a request response DLPDU (BA_WAITING_DAT or BA_WAITING_RQ status), after its emission of a variable identifier or request identifier DLPDU.
- The length of timer T1 is equal to the length of timer T0.
- Expiration of the timer gives the bus-arbitrator a new status: that of emitting the next identifier DLPDU (NEXT status).

5.6.3.2 Timer T2

- Is located in the active bus-arbitrator
- Its purpose is to monitor the length of the synchronization window of the basic cycle. The active bus-arbitrator emits the identifier DLPDUs of the basic padding sequence as long as the T2 timer has not expired.
- Is activated at the beginning of each basic synchronized cycle, when the bus-arbitrator takes on the status of emitting identifier DLPDUs (NEXT status after BA_WAITING_SYNC status).
- Is not deactivated by any event.
- The length of the synchronization window is a function of the basic cycle's aperiodic exchanges or message services;
- Expiration of the timer gives the bus-arbitrator a new status: that of emitting the next identifier DLPDU (NEXT status).

5.6.3.3 Timer T3

- Is located in potential bus-arbitrators
- Its purpose is to monitor the active bus-arbitrator's emission of identifier DLPDUs within a given time.
- Is active as long as the bus-arbitrator has potentially active status.
- The length of the T3 timer should be superior to the length of the T0 timer. It should have a different value for each of the potential bus-arbitrators. The note below suggests a value for this length.
- Expiration of the timer triggers the activation of a bus-arbitrator.

NOTE T3 timer dimensions are set as follows:

$T3 = k \times n \times T0$ with

$n > 2$ a function of the geographic address of the bus-arbitrator to which T3 is attached and
 k a coefficient that allows the bus-arbitrator's turnaround time to be covered.

5.6.3.4 Timer T4

- Is located in stations
- Its purpose is to monitor the emission of a variable response DLPDU on the bus within a given time, in order for example to correctly associate a DLCEP-identifier DLPDU received with the corresponding response DLPDU.
- Is active when the station is waiting for a variable response DLPDU (WAITING_DAT status).
- The length of T4 is equal to the length of T0

- Expiration of the timer causes the station to be placed in the status of waiting for the next identifier DLPDU (WAITING_ID status).

5.6.3.5 Timer T5

- Is located in the active bus-arbitrator
- Its purpose is to monitor a message source station's emission of a response DLPDU indicating the end of a message transaction within a given time.
- Is active when the bus-arbitrator is waiting for an end of message response DLPDU (BA_WAITING_END status), after its emission of a message identifier DLPDU.
- The length of timer T5 should be greater than the length of timer T0. The note below suggests a value for T5.
- Expiration of the timer gives the bus-arbitrator a new status: that of emitting the next identifier DLPDU (NEXT status).

NOTE The length of timer T5 is equal to two times the length of timer T0.

With such a choice of lengths collision between two operators is avoided. For example, when an acknowledgement DLPDU RP_ACK is lost the bus-arbitrator cannot emit a second ID_MSG at the same time as the source station answers with the emission of RP_MSG_ACK (see T6).

5.6.3.6 Timer T6

- Is located in stations
- Its purpose is to monitor the emission of an acknowledgement DLPDU by the destination station within a given time.
- Is active when the source station is waiting for an acknowledgement DLPDU (WAITING_ACK status) after the end of its emission of an acknowledged message response DLPDU.
- The length of T6 is equal to the length of T0
- Expiration of the timer causes the station to be placed in the status of re-emitting the message if possible (RESTART_SOURCE status) or else in the status of waiting for the next identifier DLPDU (WAITING_ID status) after emitting the end of message transaction DLPDU.

The definition of these timers is necessary to guarantee the ability to interoperate of the various devices connected to a DLL local link.

All the timers within a single local link should have the same value. A default value is defined. These values can be modified by a station's system management as long as the equivalence of values is respected. Values will be agreed upon later in accordance with implementation needs.

6 DLPDU-specific structure, encoding and element of procedure

6.1 General

This clause describes the structure, contents of the PDU and specifies elements of procedure related to the invoked service.

Behavior of particular cases is described in detail in the relative subclause by the use of dedicated state machines.

6.2 Buffer read

Buffer reading service is local to an entity since the interactions related to this service take place only through the DLS-user/DLL interface, as shown in Figure 24. Correspondence

between DLSDU service data units and interactions between the DLL and the communication medium are provided by the buffer transfer service.

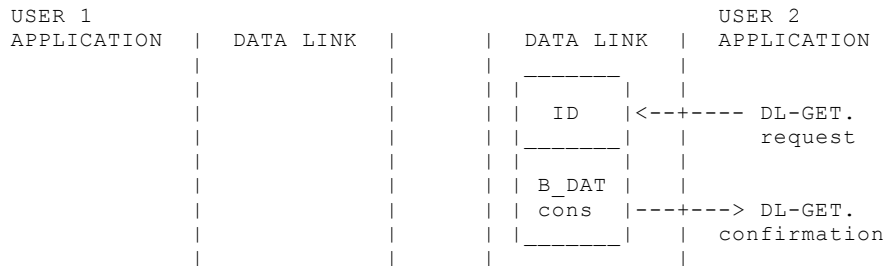


Figure 24 – Buffer reading service interactions

6.3 Buffer write

Buffer writing service is local to an entity since the interactions related to this service take place only through the DLS-user/DLL interface, as shown in Figure 25. Correspondence between DLSDU service data units and interactions between the DLL and the communication medium are provided by the buffer transfer service

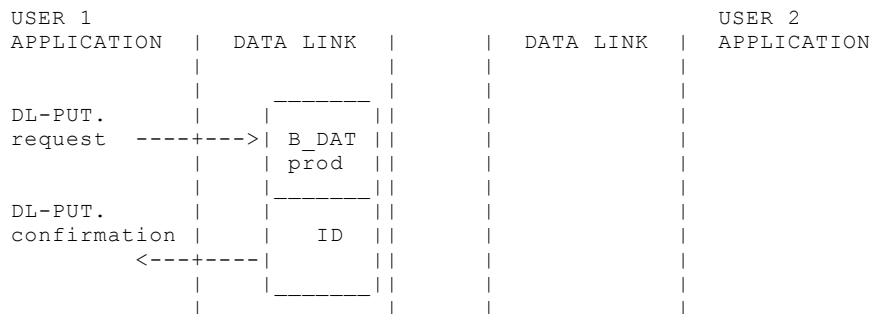


Figure 25 – Buffer writing service interactions

6.4 Buffer transfer

There are two phases to buffer transfer for all variables, as shown in Figure 26:

- the bus-arbitrator emits the identifier DLPDU associated with the variable,
- the producing entity emits the corresponding response DLPDU and generates a DL-SENT indication primitive for the DLS-user. The value of the variable is picked up by all consuming entities and a DL-RECEIVED indication primitive is generated and passed on to each of the DLS-users concerned.

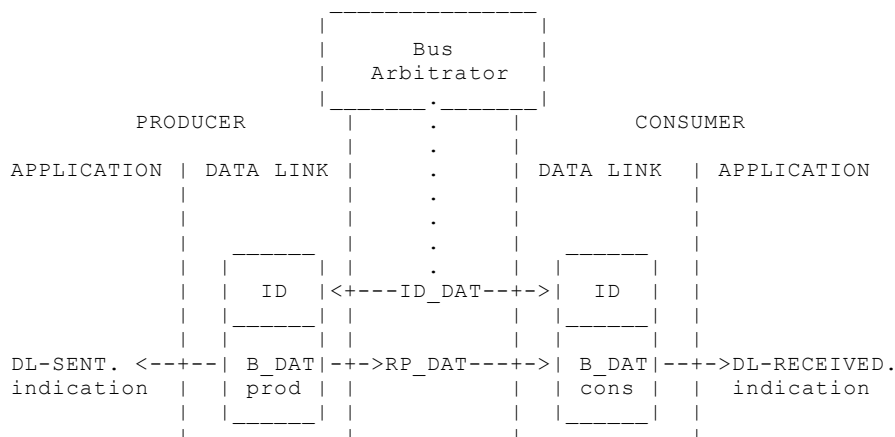


Figure 26 – Buffer transfer service interactions

6.4.1 Associated DLPDUs

A buffer transfer involves the circulation of two DLPDUs.

The bus-arbitrator broadcasts a DLCEP-identifier DLPDU (ID_DAT) that allocates the medium to the entity that produces the associated variable. The producing entity then broadcasts the variable in a response DLPDU (RP_DAT) that is picked up by the consuming entity or entities. The phases of this sequence are shown in Figure 27.

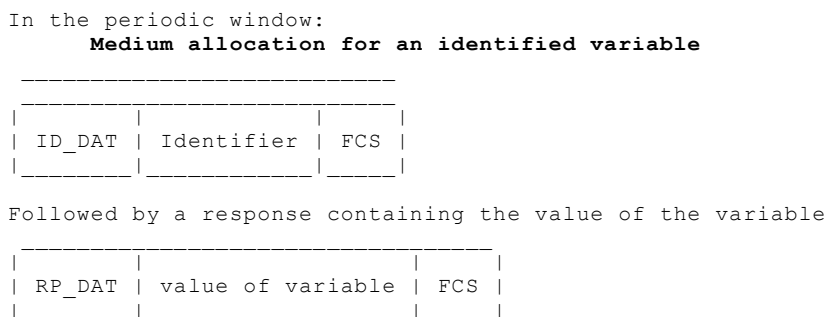


Figure 27 – Buffer transfer DLPDU sequence

NOTE A description of all DLPDUs, and the use of the various fields, can be found in 4.5.

6.5 Specified explicit request

A specified explicit request for a variable exchange can be initiated by any entity. The request triggers exchanges whose form depends on the value of the indicator RQ_INHIBIT associated with SPECIFIED_IDENTIFIER. This value is set upon configuration.

If RQ_INHIBIT = FALSE, there are three phases to the exchange:

phase 1:

Following a DL-SPEC-UPDATE request primitive, the initiating entity uses the control field of the variable response DLPDU (corresponding to the identifier specified when the service was requested) to ask the bus-arbitrator to circulate an urgent request identifier.

phase 2:

The bus-arbitrator emits the corresponding request identifier in the window allocated to triggered variable exchanges (aperiodic window),

phase 3:

The initiating entity emits a request response DLPDU whose "data" field contains the list of identifiers that the entity would like broadcast. The initiating entity also transmits a DL-SPEC-UPDATE confirm primitive to the DLS-user. One or more buffer transfers then take place within the aperiodic traffic cycle.

- the bus-arbitrator emits the identifiers of the variables requested,
- the producer of each variable requested emits a variable response DLPDU.

If RQ_INHIBIT = TRUE, there are two phases to the exchange:

phase 1:

Regardless of requests, for all identifiers set aside for specified service whose RQ_INHIBIT indicator is "true" the bus-arbitrator emits a request identifier in the periodic window.

phase 2:

The initiating entity emits a request response DLPDU whose "data" field contains the list of variable identifiers that the entity would like broadcast. The initiating entity also transmits a DL-SPEC-UPDATE confirmation primitive to the DLS-user. One or more buffer transfers then take place within the periodic traffic cycle immediately following the request.

- the bus-arbitrator emits the identifiers of the variables requested,
- the producer of each variable requested emits a variable response DLPDU.

The interaction sequences for the two classes of non-erroneous operation are shown in Figure 28 and Figure 29.

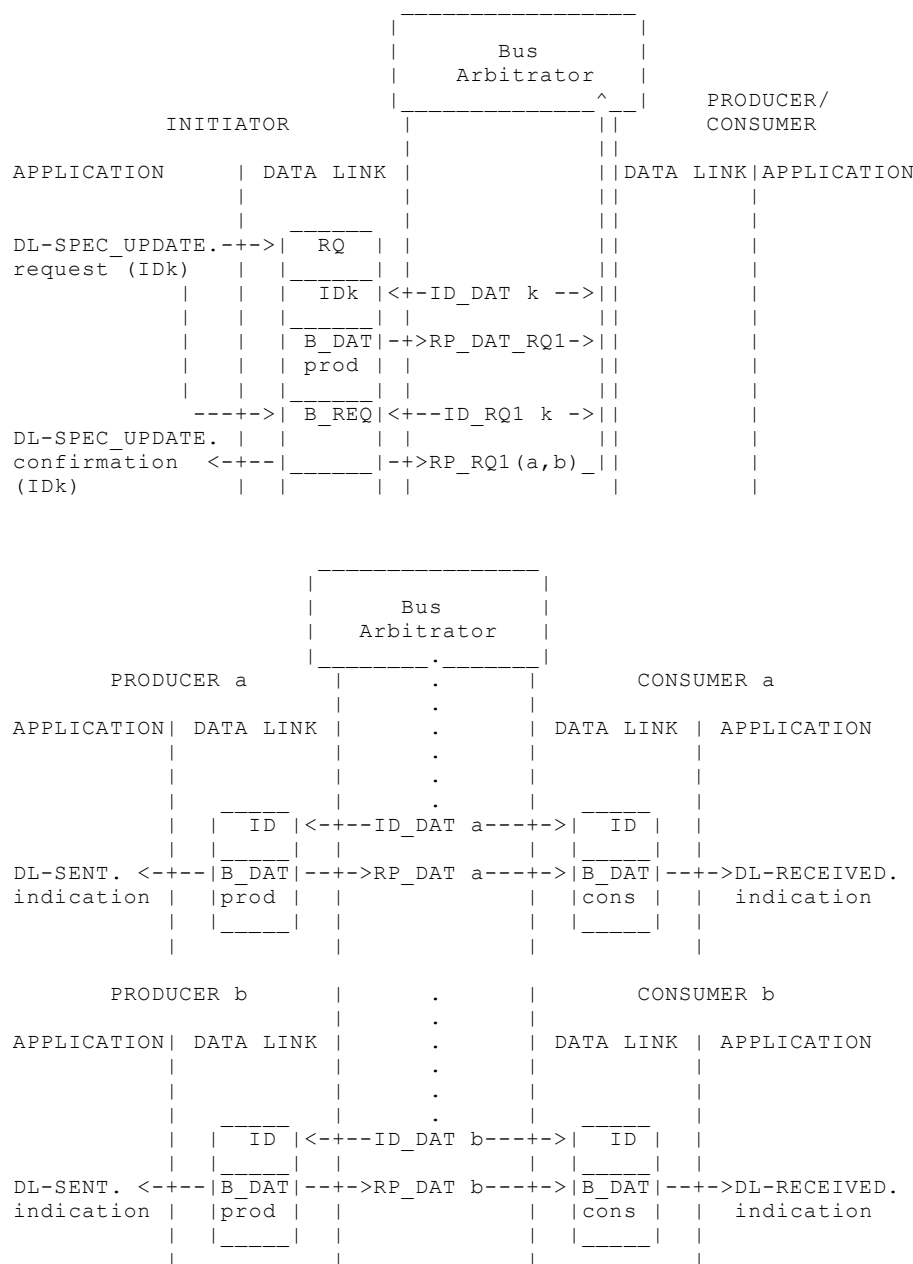


Figure 28 – Interactions within the specified explicit request for buffer transfer service in the aperiodic window

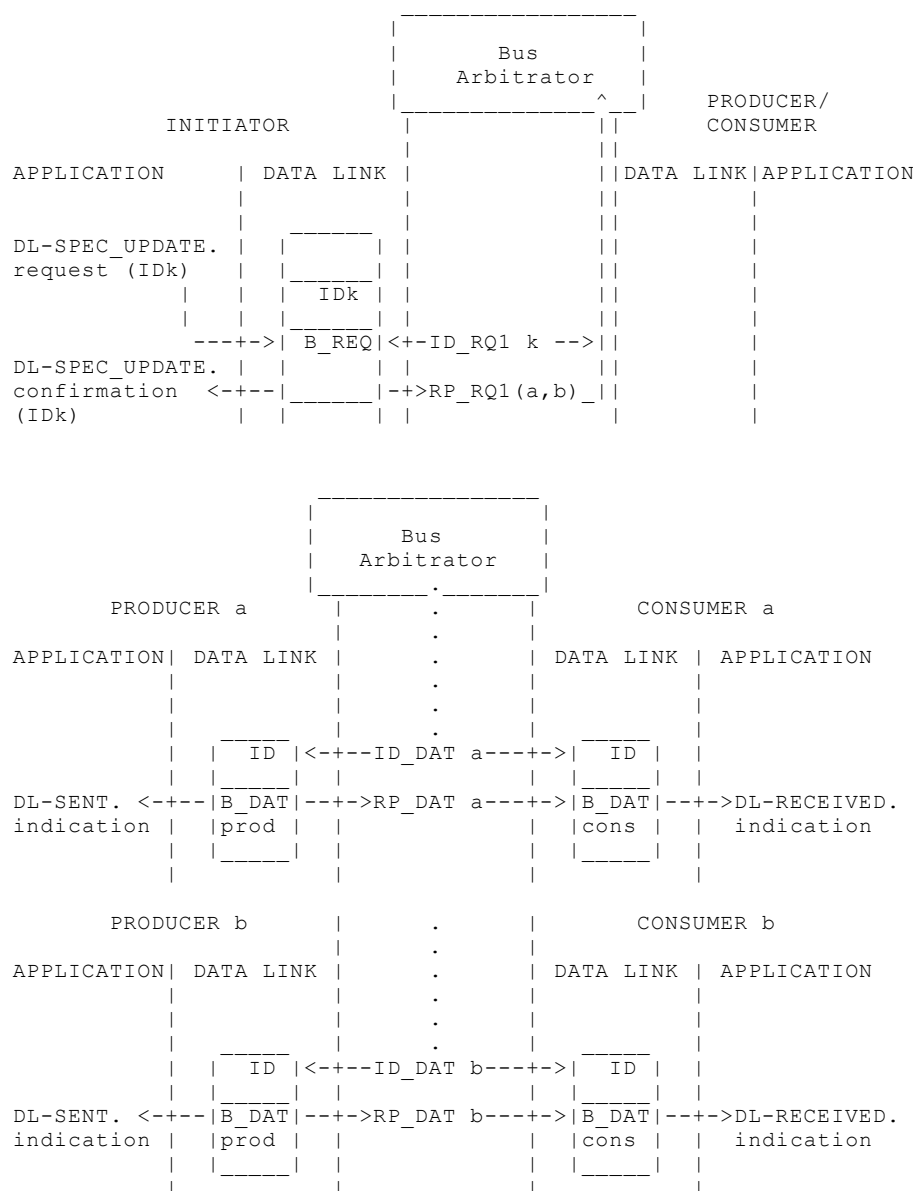


Figure 29 – Interactions within the specified explicit request for buffer transfer service in the periodic window

6.5.1 Associated DLPDUs

An explicit request for a buffer transfer corresponds to the circulation of three DLPDUs. The first of these DLPDUs is part of normal periodic scanning.

The initiating entity broadcasts a variable exchange response DLPDU whose control field includes the request to the bus-arbitrator to circulate a request identifier. The initiating entity states the priority level (RP_DAT_RQi, or RP_DAT_RQi_MSG if an aperiodic message service is also requested with this identifier).

The bus-arbitrator then broadcasts a request identifier DLPDU (ID_RQi) during an aperiodic window. The bus-arbitrator uses the identifier which carried the request.

The list of variable identifiers to be broadcast is sent to the bus-arbitrator (RP_RQi).

The bus-arbitrator responds with one or more buffer transfers: (ID_DAT, RP_DAT) transaction.

If the specified explicit request is part of the bus-arbitrator's periodic scanning cycle, the circulation of the first DLPDU is null and the other DLPDUs are exchanged during aperiodic scanning.

Figure 30 shows the DLPDU sequence for an explicit request for a transfer:

In the periodic window:

Medium allocation for an identified variable

ID_DAT	Identifier	FCS
--------	------------	-----

followed by a response with an explicit request for a transfer, and the transfer's priority:

RP_DAT_RQi	value of the variable	FCS
------------	-----------------------	-----

or

RP_DAT_RQi_MSG	value of the variable	FCS
----------------	-----------------------	-----

and then a number of identifiers later, in the aperiodic window:

**Medium allocation for an identified variable
and a response list of variable identifiers**

ID_RQi	Identifier	FCS
--------	------------	-----

followed by a list of the identifiers required

RP_RQi	list of identifiers	FCS
--------	---------------------	-----

followed by the transactions at a later time during the aperiodic window:

- medium allocation for identified variables (ID_DAT)
- associated responses (RP_DAT).

Figure 30 – DLPDU sequence for an explicit request for a transfer

6.5.2 Behavior with mismatched parameters

6.5.2.1 Buffer transfer specified explicit request (RQ_INHIBITED = False)

The evaluation network of Figure 31 allows specifying the behavior of a producer/consumer entity in the following cases:

- reception of an ID_RQ2 DLPDU containing a DLCEP-identifier bearing a buffer transfer exchange specified explicit request (RQ=1),
- reception of an ID_RQ1 DLPDU containing a DLCEP-identifier not bearing a buffer transfer specified explicit request (RQ=0).